

Universidade de Lisboa
Instituto de Geografia e Ordenamento do Território



**Desenvolvimento de uma aplicação webSIG para conteúdos
turísticos na Câmara Municipal de Machico**

Paulo Sérgio Henriques de Freitas

Relatório de Estágio orientado
por Professor Doutor Nelson Miguel Branco Mileu

Mestrado em Sistemas de Informação Geográfica e Modelação
Territorial Aplicados ao Ordenamento

2019

Universidade de Lisboa
Instituto de Geografia e Ordenamento do Território



**Desenvolvimento de uma aplicação webSIG para conteúdos
turísticos na Câmara Municipal de Machico**

Paulo Sérgio Henriques de Freitas

Relatório de Estágio orientado
por Professor Doutor Nelson Miguel Branco Mileu

Júri:

Presidente: Professor Doutor Nuno Manuel Sessarego Marques da Costa do
Instituto de Geografia e Ordenamento do Território da Universidade de
Lisboa

Vogais:

- Professor Doutor João António dos Reis da Escola Superior de Hotelaria e Turismo do Estoril
- Professor Doutor Nelson Miguel Branco Mileu do Instituto de Geografia e Ordenamento do Território da Universidade de Lisboa

2019

Agradecimentos

Deixo aqui algumas palavras para expressar a minha gratidão às pessoas e instituições que, direta ou indiretamente, me ajudaram a cumprir os meus objetivos e a realizar mais uma etapa da minha vida.

Agradeço em primeiro lugar ao meu orientador, Professor Doutor Nelson Mileu, pela disponibilidade que teve ao longo destes meses para me ajudar e esclarecer com o seu conhecimento, pelo seu apoio e acompanhamento pessoal constante.

Agradeço ao Instituto de Geografia e Ordenamento do Território por me permitir concretizar esta etapa da minha vida e ser uma instituição tão próxima dos seus alunos, e por isso ímpar.

Agradeço às várias entidades, cujo contributo foi fulcral, por me terem recebido e disponibilizado dados e informação: Câmara Municipal de Machico, em particular nas pessoas do professor Rafael da Fonseca e engenheiro Henrique Teixeira.

A todos os meus amigos e aos meus colegas da residência que me acompanharam ao longo deste ano, um obrigado por tudo. Sem vocês esta caminhada teria sido mais difícil.

Um agradecimento especial à Rafaela que esteve sempre ao meu lado desde o primeiro dia e que foi uma fonte de motivação inesgotável. Agradeço a compreensão inerente aos momentos que lhe possa ter tirado para dedicar à realização deste relatório de estágio e pela ajuda que sempre disponibilizou ao longo desta caminhada.

Por fim, mas certamente a parte mais importante, dedico um agradecimento especial à minha família, pela educação que me deu, pelos valores que me transmitiu e por todos os ensinamentos de vida que me proporcionou. À minha mãe, agradeço todo o esforço e dedicação aos seus filhos, por todos os conselhos que me deu, e por todo o amor e carinho oferecidos ao longo de uma vida, porque sem ela nada disto teria sido possível.

Resumo

Na Madeira o sector do turismo é determinante para o bom funcionamento da economia. Para tal é fulcral a administração pública disponibilizar as ferramentas necessárias para os visitantes usufruírem de uma estada confortável.

Uma das ferramentas utilizadas para a administração pública disponibilizar os pontos de interesse dos seus municípios são os webSIG, que estão em grande crescimento nos últimos anos na administração pública.

Um dos grandes problemas dos webSIG, que têm sido implementados nos municípios, prende-se com estes serem baseados em arquiteturas aplicacionais complexas. Apesar destas arquiteturas aplicacionais facilitarem o trabalho de desenvolvimento, estas reduzem a versatilidade e flexibilidade dos webSIG, o que, conseqüentemente, os torna muito onerosos e limitados em termos de evolução.

Neste contexto, o objetivo deste relatório de estágio é o desenvolvimento de um webSIG para o *website* VisitMachico, que torne mais fácil a respetiva procura dos pontos de interesse.

O desenvolvimento da aplicação é orientado pelas seguintes questões:

1. Que temas é que vamos abordar e trazer para dentro do webSIG?
2. Qual será a melhor maneira de tratar os dados (dados estes que se encontram em formato shapefile) e de converter os mesmo para introduzir dentro do webSIG?
3. Quais são as linguagens mais versáteis e que trazem uma maior flexibilidade futura à aplicação?
4. Que ferramentas é que a aplicação necessita para ser funcional e para ser *user friendly*?
5. Quais são as formas de tornar a aplicação possível de ser gerida *a posteriori* pelos funcionários da Câmara Municipal de Machico?

De maneira a cumprir com os objetivos do estágio, foram criados diversos requisitos funcionais e não funcionais, para que a aplicação fosse ao encontro do pretendido pela Câmara Municipal de Machico. Para entender os diferentes caminhos que a aplicação poderia seguir, foram analisados diferentes webSIG, em diferentes contextos, de maneira a perceber os recursos e *designs* mais favoráveis para a realidade da nossa aplicação.

Com a aplicação concluída, foi possível verificar que o JavaScript e a biblioteca Leaflet são uma boa alternativa para a construção de um webSIG, transformando a aplicação simples e *user friendly* cumprindo com todos os requisitos propostos pela Câmara Municipal de Machico, tornando assim mais fácil a procura dos pontos de interesse do Município.

Palavras-chave: webSIG; Turismo; Administração Local

Abstract

In Madeira, the tourism sector is crucial for the smooth functioning of the economy. For that, it is essential for the public administration to provide the appropriate tools necessary for visitors' comfortable visit.

One of these tools to make the points of interest available is the use of a WebGIS, a tool in great growth in recent years in public administration.

One of the major problems with WebGIS that have been implemented in municipalities is their complex architectural structures. Although these application architectures facilitate development work, they also reduce the versatility and flexibility of WebGIS, which consequently makes them very costly and limited in terms of evolution.

In this context, the objective of this report is the development of a WebGIS for the VisitMachico website that facilitate the location your points of interest.

The development of the application is guided by the following questions:

1. What themes are going to be approached and convert them to include in WebGIS?
2. What is the best way to treat and convert the data (originally in shapefile format) to insert them into the WebGIS?
3. What are the most versatile languages bringing greater flexibility to the future application?
4. What tools are required to make the application functional and user-friendly?
5. What is the best, appropriate way to make the application manageable a posteriori by the staff at the Town Hall of Machico?

In order to comply with the objectives of the internship, several functional and non-functional requirements were created to meet the Town Hall's application needs. To understand the different paths that the application could follow, different WebGIS were analysed in diverse contexts, in order to perceive the more favourable features and designs for the reality of our application.

With the application completed, it was possible to verify that JavaScript and the Leaflet library are good alternatives for the configuration of a WebGIS, making the application simpler and user-friendly, and complying with all the requirements proposed by the Town Hall of Machico to facilitate the search of the Municipality's points of interest.

Keywords: WebGIS; Tourism; Local Administration

Índice

| | |
|---|----|
| Capítulo 1 – Introdução | 1 |
| 1.1. Enquadramento do tema | 1 |
| 1.2. Entidade acolhedora do estágio | 2 |
| 1.3. Definição do problema | 3 |
| 1.4. Descrição do estágio..... | 4 |
| 1.4.1. Objetivos do estágio | 4 |
| 1.4.2. Atividades e tarefas | 4 |
| 1.4.3. Metodologia | 5 |
| 1.4.4. Utilizadores da solução..... | 6 |
| 1.5. Estrutura do relatório | 7 |
| Capítulo 2 – Revisão da literatura..... | 9 |
| 2.1. SIG como recurso turístico | 9 |
| 2.2. Aplicações de webSIG na divulgação de recursos turísticos | 10 |
| Capítulo 3 – Análise e desenho do sistema | 13 |
| 3.1. Definição do problema | 13 |
| 3.2. Análise de requisitos..... | 14 |
| 3.2.1. Quadro dos requisitos funcionais..... | 16 |
| 3.2.2. Quadro dos requisitos não funcionais..... | 17 |
| 3.3. Desenho da aplicação..... | 18 |
| Capítulo 4 – Camada de dados | 20 |
| 4.1. Recolha e origem dos dados..... | 20 |
| 4.2. Estrutura dos dados..... | 21 |
| 4.2.1. Diferentes temas com tabelas de atributos desorganizadas | 21 |
| 4.2.2. Dados com <i>shapefile</i> , mas sem descrição na tabela de atributos e dados sem <i>shapefile</i> | 22 |
| 4.2.3. Dados alfanuméricos | 23 |
| 4.3. Carregamento dos dados | 23 |
| Capítulo 5 – Implementação..... | 26 |
| 5.1. Linguagens e bibliotecas..... | 26 |
| 5.1.1. Linguagem de marcação (HTML5)..... | 26 |

| | | |
|--|---|----|
| 5.1.2. | Linguagem de estilo (CSS)..... | 27 |
| 5.1.3. | Linguagem JavaScript | 28 |
| 5.1.4. | Biblioteca Leaflet | 29 |
| 5.1.5. | Biblioteca jQuery | 29 |
| 5.2. | Desenvolvimento do <i>script</i> | 30 |
| 5.2.1 | <Head> | 30 |
| 5.2.2 | <Body> | 31 |
| 5.2.2.1 | <Script> | 31 |
| 5.2.3. | Mapas de base..... | 32 |
| 5.3. | Funcionalidades da aplicação webSIG..... | 32 |
| 5.3.1. | Descrição dos vários mapas..... | 34 |
| 5.4. | Casos de estudo..... | 37 |
| Capítulo 6 – Resultados e conclusões..... | | 40 |
| 6.1. | Alteração da aplicação a posteriori | 40 |
| 6.2. | Integração da aplicação webSIG no <i>website</i> VisitMachico | 42 |
| 6.2.1. | Possíveis alterações da aplicação | 44 |
| 6.3. | Síntese e conclusões do estágio | 44 |
| Bibliografia..... | | 46 |
| Apêndice A: Script da aplicação (Mapa Ecosistemas) | | 48 |
| Apêndice B: Script da aplicação (Mapa mar)..... | | 52 |
| Apêndice C: Script da aplicação (Mapa miradouros) | | 57 |
| Apêndice D: Script da aplicação (Mapa museus e arte urbana)..... | | 61 |
| Apêndice E: Script da aplicação (Mapa património municipal)..... | | 66 |
| Apêndice F: Exemplo de um GeoJSON | | 71 |

Índice de figuras

| | |
|---|----|
| Figura 1: Freguesias do Município de Machico | 3 |
| Figura 2: Fases de desenvolvimento da aplicação. | 6 |
| Figura 3: Fluxograma da estrutura do relatório. | 8 |
| Figura 4: webSIG sobre o turismo de Ourém. Fonte: http://websig.cm-ourém.pt/turismo/index.php?page=mapa . Acesso em: 10 junho 2019. | 10 |
| Figura 5: Aplicação Lisboa interativa. Fonte: http://lxi.cm-lisboa.pt/lxi/ Acesso em: 11 junho 2019. | 11 |
| Figura 6: Aplicação Sygic travel. Fonte: https://maps.sygic.com/?utm_source=content-pages&utm_medium=cta&utm_campaign=homepage#/?map=13,38.7174,-9.1321 Acesso em: 13 junho 2019. | 12 |
| Figura 7: Mapa do website Visit Machico. Fonte: https://www.visitmachico.com/pt/sentir/ecossitios/encontre-o-seu-ecossitio Acesso em: 17 junho 2019. | 14 |
| Figura 8: Camada de aplicação. | 15 |
| Figura 9: Esboço da aplicação. | 18 |
| Figura 10: Esquema dos campos do toggle. | 20 |
| Figura 11: Diferentes shapefiles e com tabelas de atributos desorganizadas. | 21 |
| Figura 12: Dados com shapefile, mas sem descrição na tabela de atributos. | 22 |
| Figura 13: Conversão para GeoJSON. | 24 |
| Figura 14: Ligação do GeoJSON com o script. | 25 |
| Figura 15: Obtenção de informação relativa à Estátua de Tristão Vaz Teixeira. | 33 |
| Figura 16: Layout de abertura do mapa Ecosítios. | 33 |
| Figura 17: Zoom in após pesquisa do Miradouro da Quinta do Lorde, na versão móvel. ... | 34 |
| Figura 18: Mapa com o tema Mar. | 34 |
| Figura 19: Mapa com o tema Miradouros. | 35 |
| Figura 20: Mapa com o tema Património municipal. | 35 |
| Figura 21: Mapa com o tema Ecosítios. | 36 |
| Figura 22: Mapa com o tema Museus e Arte. | 36 |
| Figura 23: Ícone com mudança de cor, após a passagem do rato. | 37 |
| Figura 24: Toggle ativo e imagem de satélite. | 38 |
| Figura 25: Mapa Ecosítios com destaque da barra de pesquisa. | 38 |
| Figura 26: Zoom out máximo do mapa Ecosítio. | 39 |
| Figura 27: Versão móvel do mapa Museus e Arte | 39 |
| Figura 28: Alteração do zoom e da janela de abertura | 40 |
| Figura 29: Alterar ícones. | 41 |
| Figura 30: Alterar a Barra de pesquisa. | 41 |
| Figura 31: Alterações no GeoJSON. | 42 |
| Figura 32: WebSite VisitMachico, após ser implementada a aplicação. | 43 |

Índice de quadros

| | |
|--|----|
| Quadro 1: Requisitos funcionais..... | 16 |
| Quadro 2: Requisitos não funcionais..... | 17 |

Acrónimos e Abreviaturas

AJAX- *Asynchronous JavaScript and XML*

API- *Application programming interface.*

CERN- *European Organization for Nuclear Research (Conseil Européen pour la Recherche Nucléaire)*

CMM- *Câmara Municipal de Machico.*

CSS- *Cascading Style Sheets.*

ECMA- *European Computer Manufacturers Association*

Esri- *Environmental Systems Research Institute*

GIS- *Geographic information system.*

HTML- *HyperText Markup Language.*

HTTP- *Hypertext Transfer Protocol.*

RAM - *Região Autónoma da Madeira.*

SHP- *Shapefile.*

SIG- *Sistemas de Informação geográfica.*

SIGMTO- *Sistemas de Informação Geográfica e Modelação Territorial Aplicados ao Ordenamento.*

SQL- *Structured Query Language.*

webSIG- *Sistemas de Informação geográfica em rede.*

WMS - *Web Map Service.*

WWW- *Word Wide Web.*

W3C- *World Wide Web Consortium.*

Capítulo 1 – Introdução

1.1. Enquadramento do tema

Com a criação da World Wide Web (WWW) em 1989, por Tim Berners-Lee, a transferência de informação e comunicação foi alterada drasticamente.

Atualmente, perante a carência de uma resposta e a subsequente procura por informação, recorremos à internet, que consegue prover uma quantidade infinda de informação instantaneamente. A maneira que os sistemas de informação geográfica (SIG), e por sua vez a geografia, tiveram de se difundir e de comunicar com o mundo da Word Wide Web foi através dos webSIG.

O webSIG surge da necessidade de disponibilizarmos informação espacial de forma rápida e fácil, de modo a que qualquer pessoa consiga aceder à mesma. Infelizmente não é possível para qualquer pessoa aceder a um software de SIG, por exemplo por não ter tempo para usar o mesmo com eficiência. Para colmatar este défice ao acesso de informação geográfica surgem os webSIG que trouxeram uma forma fácil de disseminar informação espacial na internet.

Nos tempos que correm qualquer entidade pública ou entidade privada que não esteja referenciada num mapa online é uma entidade sem presença física/espacial no mundo real, salvo raras exceções. É uma preocupação primária para qualquer entidade georreferenciar os seus pontos de interesse, de modo que a sua informação seja disponibilizada de forma representativa no espaço e visualmente mais apelativa a quem procura a mesma (Alesheikh et al., 2015).

Ciente da importância que a informação espacial tem para o turismo de uma região, a Câmara Municipal de Machico, em conjunto com a equipa do *website* Visit Machico, *website* este com conteúdos unicamente turísticos, resolveu melhorar a representação espacial dos pontos de interesse no *website* Visit Machico através do desenvolvimento de uma aplicação webSIG unicamente focada em conteúdos turísticos e na sua divulgação.

Este relatório de estágio tem como principal objetivo desenvolver um webSIG para o *website* Visit Machico com recurso a linguagem de programação JavaScript. Ao longo deste projeto foi ainda utilizada a linguagem de marcação HTML, a linguagem de estilo CSS e as bibliotecas Leaflet e jQuery.

“I hope we will use the Net to cross barriers and connect cultures”

Tim Berners-Lee

1.2. Entidade acolhedora do estágio

O presente relatório teve implícito a realização de um estágio curricular na Câmara Municipal de Machico (CMM), sendo que o mesmo decorreu nos meses de março de 2019 a julho de 2019. A realização do estágio teve como fim a obtenção do grau de Mestre em Sistemas de Informação Geográfica e Modelação Territorial Aplicados ao Ordenamento (SIGMTO).

Nestes cinco meses de estágio cooperei num projeto do Departamento de Urbanismo da CMM, intitulado VisitMachico, tendo como orientador na entidade acolhedora o Professor Rafael Oliveira da Fonseca. Este projeto tem como principal objetivo a promoção do município e de toda a oferta turística existente no mesmo, sendo que toda esta informação terá de ser difundida de maneira interativa e dinâmica através da esfera digital VisitMachico®.

Para além de cooperar com o projeto VisitMachico também realizei diversas tarefas no departamento de urbanismo da CMM, como organização de bases de dados geográficas, reorganização das shapefiles da CMM, atualização do inventário da rede viária da CMM e procedi ao levantamento de alguns pontos de interesse, sendo que este último ponto já se encontra ligado ao projeto VisitMachico.

O município de Machico localiza-se na ponta Este da Região Autónoma da Madeira (RAM) e integra 5 freguesias, sendo estas Água de Pena, Caniçal, Machico, Santo António da Serra e Porto da Cruz. Este município é confinado a Noroeste por Santana, a Sudoeste por Santa Cruz e a Oeste por uma pequena faixa de território pelo município do Funchal (Figura 1).

O Município de Machico tem 68,31 km² e 21,828 habitantes (segundo os censos 2011), sendo a freguesia com mais população, a freguesia de Machico (11,947 habitantes) e com menor população a freguesia de Santo António da Serra (1,355 habitantes).

Apesar de ser, essencialmente uma área de costa, banhado a norte, este e sul pelo oceano Atlântico, a sua paisagem é marcada por uma grande diversidade de florestas densas, zonas com vegetação baixa e zonas com vegetação quase inexistente. No que diz respeito à morfologia, esta é marcada por vários montes e serras, sendo estas, a do Castanho (589 m), a do Pedreiro (792 m), Pico da Coroa (738 m) e Penha de Águia (590 m).



Figura 1: Freguesias do Município de Machico

1.3. Definição do problema

Os sistemas de informação geográfica são atualmente um dos principais pilares da modernização administrativa dos municípios, sendo estes fundamentais para a análise, gestão e implementação de qualquer estratégia nos mesmos. Deste modo, com este trabalho pretendemos implementar um webSIG para o Município de Machico, com o objetivo de disponibilizar de maneira mais simples a informação geográfica aos colaboradores do município, munícipes e visitantes.

A grande panóplia de ferramentas e a interface pouco *“user friendly”*, torna os sistemas de informação geográfica pouco intuitivos. Devido a essa problemática e à dificuldade de acesso a essa informação, a equipa do *website* VisitMachico decidiu criar um webSIG que permitisse o acesso a informação geográfica de uma forma simples, utilizando apenas um browser e sem necessidade de instalar um software dedicado a SIG.

O município de Machico embora seja dos municípios com mais população na ilha da Madeira, infraestruturas e pontos de interesse, tem algumas dificuldades em atrair e atrair turistas e aumentar a permanência ou estada (DREM, 2017). Este município é, portanto, um ponto de passagem, mas não uma escolha óbvia para pernoitar. Consequentemente, o município não está a tirar o real valor de um sector tão importante para a região autónoma da madeira como é o turismo.

Deste modo, o webSIG que será implementado terá uma vertente essencialmente turística, com uma detalhada descrição de pontos de interesse, possibilitando assim que o turista saiba, através de uma única página tudo o que o município tem para oferecer, sendo aliciado a permanecer mais tempo no Município. No *website* Visit Machico podemos encontrar já algumas representações geográficas, mas estas são pouco dinâmicas e

atraentes para quem visita o *website*, faltando muitas ferramentas básicas para este corresponder às necessidades dos potenciais visitantes do *website* e do município.

1.4. Descrição do estágio

1.4.1. Objetivos do estágio

O objetivo principal do estágio passa por desenvolver um webSIG para divulgação de conteúdos turísticos. Este webSIG deve ser dinâmico, de fácil interpretação e apelativo na ótica do utilizador, cumprindo-se assim as finalidades de eficiente cativação e informação do utilizador.

Segue-se a enumeração dos objetivos específicos deste estágio:

- Identificar o que um turista quer ver quando chega a um novo destino;
- Realizar uma triagem dos dados disponibilizados, de modo a que o mapa não seja demasiado exaustivo visualmente e para que nele só constem os pontos de real interesse turístico;
- Converter todos os dados para o formato GeoJSON uma vez que este formato é versátil e suportado pela maioria dos SIG e APIs no mercado;
- Desenvolvimento de um webSIG de utilização “user friendly”, que possua todas as ferramentas para uma navegação eficiente e simples;
- Desenvolvimento de um *script* simples com intuito de possibilitar que, posteriormente, qualquer técnico da câmara consiga realizar alterações ou modificar dados;
- Desenvolver um *script* responsivo em qualquer tipo de plataforma;
- Realizar testes de performance à aplicação.

1.4.2. Atividades e tarefas

As atividades e tarefas desenvolvidas no decurso do estágio foram as seguintes:

- Análise de requisitos em parceria com a equipa do *website* Visit Machico;
- Conceção e idealização da estrutura do webSIG, nomeadamente Tabelas de atributos, que linguagens e bibliotecas usar, layout do webSIG, entre outros.
- Inventariação de todos os pontos, com os devidos nomes, descrições e informações necessárias sobre os mesmos;
- Elaboração e agregação das *shapefiles* que serão disponibilizadas;
- Alterar os dados do formato SHP para GeoJSON;
- Criação técnica do webSIG;
- Verificação do funcionamento do webSIG;

Simplificar o *script* de modo a que a manutenção e atualização do webSIG possa ser realizada por um técnico do município.

1.4.3. Metodologia

Considerando o objetivo de desenvolver uma aplicação webSIG para conteúdos turísticos na CMM, foi necessário dividir a realização do projeto em cinco fases (Figura 2).

A Primeira Fase corresponde à necessidade de fazer uma revisão bibliográfica sobre SIG e sobretudo sobre os webSIG, de forma a compreender o estado da arte da questão, e perceber quais as opções existentes e com melhor aplicação no projeto.

Ainda na primeira fase foi necessário compreender como colocar os dados na aplicação e qual seria o formato mais aconselhado. Depois de compreender como introduzimos os dados na aplicação prosseguimos para as linguagens e bibliotecas que iremos usar no projeto, tentando procurar as mais versáteis para este tipo de aplicação. O último passo desta fase consiste em perceber como iremos conectar de maneira fluída o webSIG com o turismo, analisando outros projetos similares e bibliografia relevante.

A segunda fase será inteiramente sobre dados, nesta fase iremos fazer uma triagem dos dados com a equipa do Visit Machico para decidir quais serão os dados utilizados. Depois de saber quais os dados mais relevantes centramo-nos na parte de organização dos mesmos, de maneira a obter uma visão geral de como ficarão na aplicação. Nesta fase iremos ainda escolher o formato dos dados. Esta é uma etapa especialmente relevante uma vez que este formato ir-nos-á acompanhar até ao final do projeto e será através do mesmo que o projeto poderá ser atualizado mais tarde. Por esta razão é importante escolher um formato simples, que cumpra o objetivo de ser atualizável por um técnico com menos experiência na área.

Por fim, trataremos os dados de maneira a que todos entrem com a mesma organização no projeto, ou seja, com a mesma tabela de atributos. Assim, criamos uma solução com menos “ruído” e mais organizada.

A terceira fase irá ser essencialmente baseada na experiência que tivemos na análise de anteriores projetos similares e bibliografia. Pois a conceptualização advém da experiência visual e dinâmica que tivemos anteriormente. A conceptualização é a altura de definir e assentar as linguagens de programação que iremos usar em todo o projeto assim como as bibliotecas e os comandos que queremos para a nossa solução.

Só na quarta fase é que começaremos realmente a construir a aplicação. Nesta fase criaremos os scripts, nas linguagens escolhidas anteriormente, escolhemos a organização/posição do mapa no *website* através do HTML, fazemos a ligação dos dados ao nosso HTML, estilizamos a página através do CSS e colocamos toda a aplicação responsiva,

para que esta possa ser utilizada tanto em *desktop*, *tablet* e *smartphone*, tanto no modo visualização vertical como na horizontal.

Na quinta e última fase serão apresentados os resultados, as devidas verificações e testes à aplicação. Nesta fase, verificamos se a aplicação é realmente simples e apelativa, tal como foi proposto nos nossos objetivos iniciais. Iremos verificar se existe algum tipo de erro, identificando-o atempadamente e fazendo as alterações necessárias, de modo a não prejudicar a navegabilidade do utilizador.

O próximo passo será verificar se a aplicação está realmente responsiva em todas as plataformas que queremos. Por último, iremos garantir que a aplicação é facilmente atualizada por uma pessoa com menos experiência. Quanto a este último aspeto mencionado, o objetivo não será que o técnico com menos experiência faça alterações significativas na aplicação, mas sim que o técnico perceba como atualizar os dados de maneira simples, ou até criar mapas através do *template* deste projeto.




| 1ª REVISÃO BIBLIOGRÁFICA | 2ª TRATAMENTO DOS DADOS | 3ª CONCEPTUALIZAÇÃO DO SISTEMA | 4ª DESENVOLVIMENTO DA APLICAÇÃO | 5ª RESULTADOS |
|--|--|---|--|--|
|  <ul style="list-style-type: none"> Bibliografia relevante sobre WebSIG. Compreender as possíveis maneiras de tratar dos dados. Quais são as linguagens mais usadas. Quais são as bibliotecas que podemos utilizar. Perceber a melhor ligação entre o WebSIG e o turismo. |  <ul style="list-style-type: none"> Triagem dos dados. Organização da base de dados. Escolha do formato dos dados. Tratamento dos dados. |  <ul style="list-style-type: none"> Desenhar o sistema. Escolher os comandos que o WebSIG terá. Definir a linguagem que vai ser utilizada no projecto. Definir as bibliotecas que vamos usar. |  <ul style="list-style-type: none"> Criação dos Scripts. Escolher a organização do Mapa. Colocar os dados na aplicação. Estilizar a página. Colocar toda a aplicação responsiva. |  <ul style="list-style-type: none"> Verificar se a aplicação está simples e apelativa. Verificar se não existe erros na aplicação. Verificar se a aplicação é responsiva. Garantir a fácil atualização da aplicação. |

Figura 2: Fases de desenvolvimento da aplicação.

1.4.4. Utilizadores da solução

Os utilizadores da solução serão todas as pessoas interessadas em conhecer melhor o município de Machico, queiram estes visitar, apenas recolher informação ou perceber a posição de determinado ponto de interesse na geografia da Madeira. Apesar de esta solução ser desenhada para todas as pessoas que queiram recolher este tipo de informação, o público-alvo da solução é claramente o turista que deseja mais informação acerca dos

pontos de interesse, de forma a proporcionar uma melhor e mais duradoura estada no município.

Evidentemente, este nicho de mercado de utilizadores da solução webSIG é coincidente com os utilizadores do *website* Visit Machico. Este *website* tem como princípio base ser uma plataforma dinâmica e em constante evolução. Para tal deve consistir numa representação geográfica avançada, visto que o utilizador deste tipo de conteúdo não procura um simples mapa.

O utilizador pode optar por visualizar o conteúdo de diferentes perspetivas, já que atualmente o trabalho de realizar um mapa não consiste em adivinhar qual é a informação que o utilizador quer observar, mas antes fazer com que o utilizador escolha como quer visualizar (Mitchell, 2005).

1.5. Estrutura do relatório

A estrutura do relatório irá assentar em seis capítulos, que passo a enumerar: 1º Introdução, 2º Revisão literária, 3º Análise e desenho do sistema, 4º Camada de dados, 5º Implementação e 6º Conclusão (Figura 3).

O relatório de estágio começa com um capítulo introdutório. O capítulo subdivide-se em cinco pontos principais, sendo estes os seguintes: Enquadramento do tema, Entidade acolhedora do estágio, Definição do problema, Descrição do estágio, onde se incluem os objetivos, tarefas, metodologia e utilizadores e, por último, a explicação da estrutura do relatório (ponto atual).

O segundo capítulo aborda os SIGs como recurso turístico e outros casos de relevo de aplicações de webSIG, para perceber que caminhos são possíveis adotar na construção da aplicação.

O terceiro capítulo será dedicado à análise e desenho do sistema. Neste tópico iremos verificar os problemas da solução anteriormente implementada no *website* Visit Machico. Faremos a análise de requisitos, posteriormente confirmada com a equipa do *website* permitindo que a solução corresponda às suas expectativas. Com estas variáveis conseguiremos desenhar o sistema, ou seja, conceptualizar o projeto. Este capítulo divide-se em: Definição do problema, análise de requisitos (funcionais e não funcionais) e desenho da aplicação.

Um capítulo é exclusivamente dedicado à análise da camada de dados, visto que é uma das partes chave do projeto. É precisamente no quarto capítulo que se dá a transferência da informação, que anteriormente estava no sistema SIG, sendo esta mesma informação disponibilizada online através de um sistema webSIG. Segue-se a subdivisão deste capítulo: Recolha e origem dos dados, estrutura dos dados, ou seja, como estes estavam dispostos *a priori* na visualização no ambiente SIG (como estavam na base de

dados) e, por fim, tratamento e carregamento destes mesmos dados devidamente transformados no formato escolhido.

O quinto capítulo consiste na implementação. Este capítulo é inequivocamente o mais longo em todo o processo, tendo em conta o tempo despendido e o volume de trabalho, devido ao facto de não dominar totalmente os fundamentos deste capítulo. Nesta fase do relatório abordaremos a linguagem de programação utilizada, as bibliotecas escolhidas, o estilo da aplicação e a sua interface.

Por último, teremos o sexto capítulo que depreende sobre os resultados. Neste capítulo são apresentados os diferentes resultados dos casos em que o projeto foi integrado no *website*. Neste capítulo está ainda presente uma análise dos resultados obtidos.

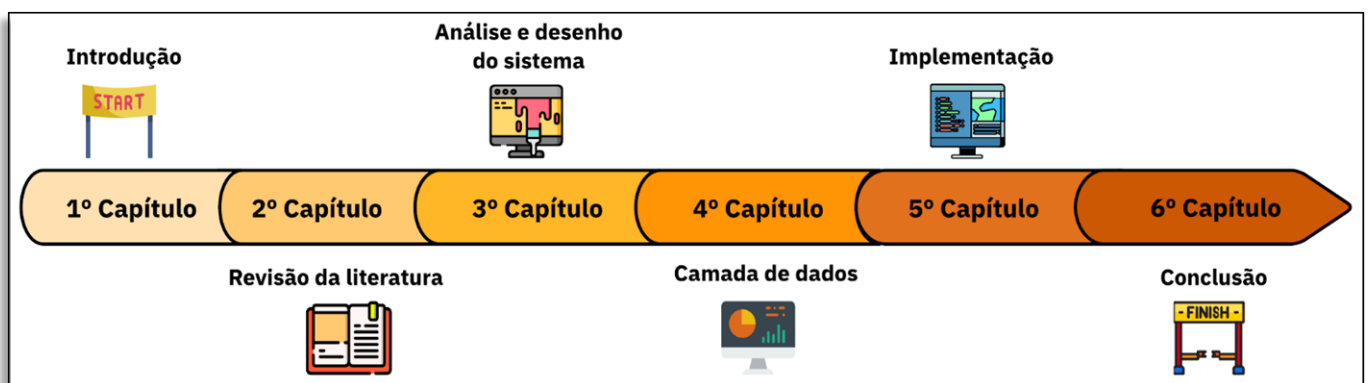


Figura 3: Fluxograma da estrutura do relatório.

Capítulo 2 – Revisão da literatura

2.1. SIG como recurso turístico

A parceria entre os SIG e o turismo é cada vez mais frequente nos dias de hoje, devido a este necessitar cada vez mais de gerir grandes volumes de informação espacial, de ser mais eficiente ou simplesmente de informar os interessados. Assim sendo, os SIG tornam-se um adequado recurso para tomar uma decisão acertada e rápida, facilitando assim quem tem a última palavra.

Durante os últimos anos, os sites de turismo ou organizações de turismo que utilizam os SIG têm aumentado exponencialmente. Atualmente qualquer tipo de *website* ou organização turística que não possua um mapa interativo está desatualizada. Tendo em conta este incremento da utilização de SIG para este fim, o aumento de software e de empresas dedicadas unicamente à união do turismo com os SIG também tem tido um aumento substancial. A indústria do turismo aplica atualmente os SIG de diversas maneiras: em mapas interativos, quiosques interativos, mapas de caminhadas 3D, análise do utilizador atual e do utilizador potencial do destino, entre outras aplicações (Duran, Shrestha & Seker, 2004).

Para a realização de uma escolha informada por parte do turista é necessária muita informação de vários campos de atividade, este desafio é dos mais complexos na conceptualização de um sistema webSIG para esta área de atividade, pois este terá que conter muita informação e de fazer sentido à primeira vista sem grande esforço por parte do utilizador (Wilkinson, Gu, Ryan, citado em Masron, Marzuki, Mohamed, & Ayob, 2014).

A informação mapeada apresentada na internet é um dos grandes fatores de decisão para o turista escolher um determinado destino, mas também para este fazer as melhores decisões quando chega ao mesmo. É neste ponto que a entidade pública é relevante na promoção do seu território, uma vez que esta terá de promover o seu município de forma a transmitir os melhores pontos de interesse aos seus turistas, de modo a que não percam tempo em locais de menor interesse.

Sem embargo dos métodos tradicionais, como publicações, brochuras, rádio e televisão serem regularmente aplicados para atingir o público alvo, estes revelam algumas restrições técnicas e de comunicação em comparação aos webSIG, que disponibilizam um sistema mais apelativo e com objetivo de satisfazer as necessidades de informação que o utilizador procura. Este sistema é mais efetivo, focalizado e mais apelativo visualmente devido à sua natureza dinâmica e interativa. Assim sendo os webSIG são uma grande mais valia para um *marketing* mais eficiente, tanto para quem procura como para quem disponibiliza este tipo de informação (Oliveira, 2015).

2.2. Aplicações de webSIG na divulgação de recursos turísticos

Existem diversos projetos de webSIG na divulgação de recursos turísticos, mas neste ponto serão apresentados três projetos diferentes. Infelizmente não foi possível analisar nenhum webSIG de outra Câmara Municipal da RAM, devido a nenhuma Câmara possuir uma aplicação deste género. Os exemplos a apresentar são: (i) um município de pequena dimensão (Ourém) com um projeto unicamente dedicado ao turismo, (ii) um grande município com um projeto não dedicado unicamente ao turismo (Lisboa) e (iii) um projeto com um perfil global exclusivamente dedicado ao turismo (Sygic travel).

O município de Ourém, apesar de ser relativamente pequeno, com cerca de 45.932 habitantes, tem uma área dedicada aos SIG com muita qualidade. Este *website* apresenta uma vasta informação SIG, desde webSIG do PDM, educação, sinalização, turismo, entre outro tipo de informações. O webSIG sobre o qual vamos debruçar e analisar é o de turismo (Figura 4). Este apresenta diversas informações, tais como onde ir, onde dormir, onde comer, património natural, locais religiosos (um dos grandes pontos de interesse deste município), cultura e património. Toda esta informação é acompanhada por diferentes ferramentas de ajuda à navegação, como botões de zoom, possibilidade de mudar para imagem de satélite e apresenta ainda um popup sempre que cliquemos em algum ícone, onde apresenta diversa informação como, nome, morada ou descrição, tornando este mapa muito interativo e intuitivo de utilizar.

Não obstante o ambiente *desktop* deste webSIG estar bem conseguido, este não é totalmente responsivo a outras plataformas, apresentando alguns erros na visualização de algumas ferramentas, como é o caso da opção de mudar para imagem de satélite, que não fica bem enquadrada no ecrã.

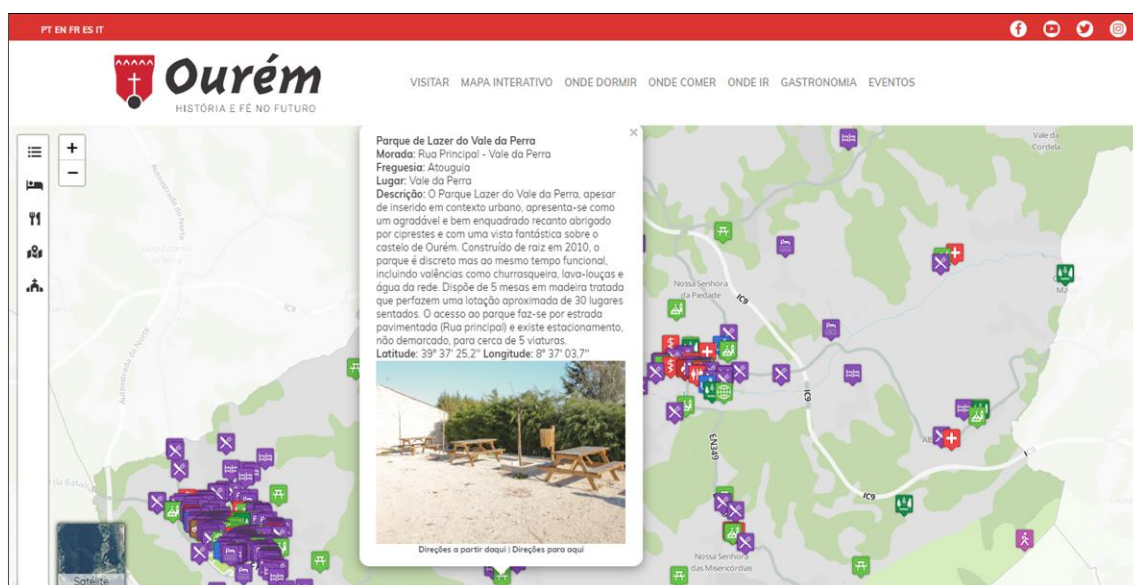


Figura 4: webSIG sobre o turismo de Ourém. Fonte: <http://websig.cm-ourem.pt/turismo/index.php?page=mapa>. Acesso em: 10 junho 2019.

O município de Lisboa possui um webSIG unicamente de uso *desktop* (Figura 5) com uma complexidade mais elevada no que diz respeito a sua elaboração e com muita informação. Este mapa interativo conhecido como Lisboa Interativa, foi criado para utilização interna e externa, tem informação sobre diversas áreas, sendo algumas delas, pontos de interesse, temática, urbanismo, na minha rua, entre outras. A área que vamos analisar incide sobre os pontos de interesse, atendendo ao facto de esta disponibilizar mais informação útil a quem queira visitar o território pela primeira vez. Devemos frisar que esta aplicação não foi construída unicamente a pensar nos turistas, consequentemente não está otimizada para esta temática.

Esta aplicação detém todas as ferramentas necessárias para uma eficaz navegação e inclui toda a informação possível numa aplicação deste género. Contudo, no que concerne à organização e à navegação intuitiva esta aplicação possui alguns problemas, visto que a vertente mais técnica desta aplicação e a enorme quantidade de informação tornam este tipo de aplicação pouco utilizada por quem quer conhecer a cidade.

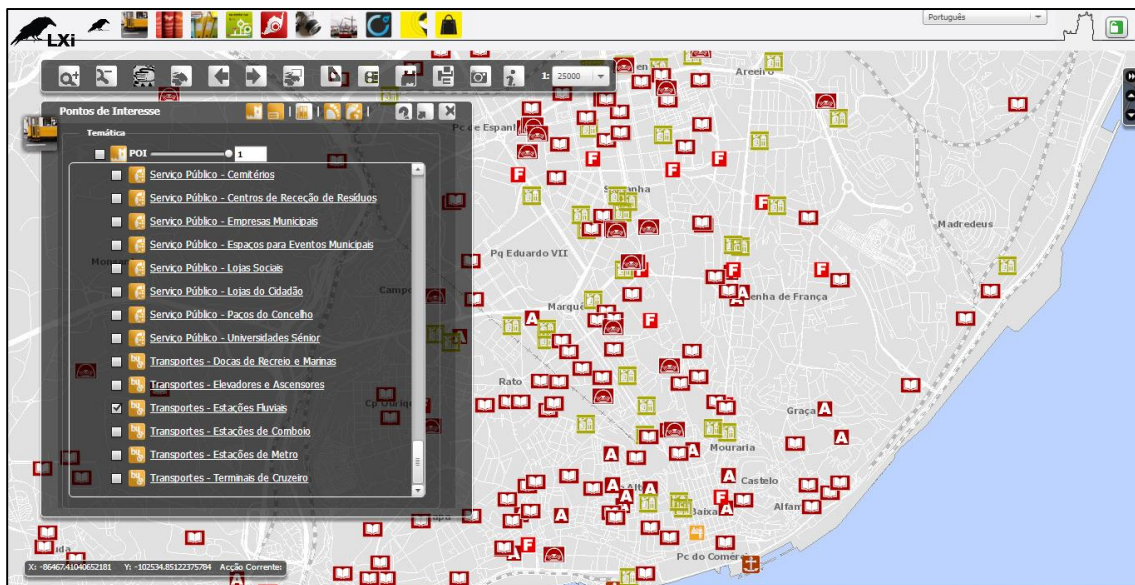


Figura 5: Aplicação Lisboa interativa. Fonte: <http://lxi.cm-lisboa.pt/lxi/> Acesso em: 11 junho 2019.

A Sygic travel (Figura 6) é um projeto unicamente pensado para quem quer conhecer o espaço que está a visitar com a ajuda de um mapa com grande volume de informação sobre atrações, hotéis, restaurantes, e outras informações relacionadas com o turismo. A Sygic travel detém todas as ferramentas para uma boa navegação. Exemplos disso são o zoom, a opção de mudar para ortofotomapa, a barra de pesquisa e escala. Para além de boas ferramentas de navegação a Sygic travel disponibiliza informação detalhada sobre a maioria dos pontos de interesse da cidade. Mais ainda, toda esta informação está traduzida em vinte e um idiomas diferentes e surge acompanhada por fotografias dos locais, opções de atividades perto do local, opiniões de visitantes, possibilidade de criar um roteiro para a nossa viagem, ligação com a booking.com, entre outras vantagens.

Outra mais valia e, simultaneamente, uma das razões para a Sygic travel ter um volume de informação tão extenso consubstancia-se no facto deste projeto ser open source

a publicações. Isto significa que qualquer utilizador pode acrescentar informação sobre os pontos de interesse. O Sygic travel também dispõe de uma aplicação móvel, o que torna este produto muito mais completo. Convém também referir que embora a aplicação seja muito completa, quase todos os serviços disponibilizados são pagos.

Depois da análise feita anteriormente, verificamos que o Sygic travel é, inquestionavelmente o pináculo da tecnologia webSIG no turismo na atualidade, este dispõe de todo o tipo de informação necessária para uma viagem informada e para uma utilização “*user friendly*”.

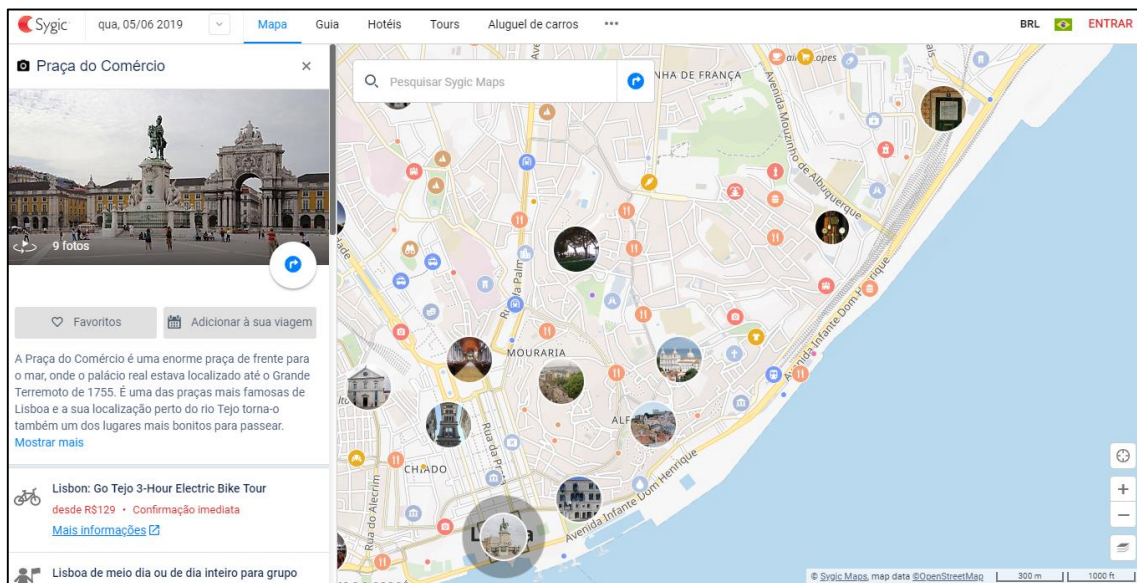


Figura 6: Aplicação Sygic travel. Fonte: https://maps.sygic.com/?utm_source=content-pages&utm_medium=cta&utm_campaign=homepage#/?map=13,38.7174,-9.1321 Acesso em: 13 junho 2019.

Capítulo 3 – Análise e desenho do sistema

3.1. Definição do problema

Não obstante o *website* Visit Machico ser relativamente recente (2017), consideramos que este está bem consolidado e contém muita informação sobre o município e o seu turismo. Contudo, no que diz respeito à informação geográfica, esta é disponibilizada de maneira pouco dinâmica e clara. Partindo desta conclusão, procuramos diagnosticar os principais problemas que tinham por consequência a experiência ser pouco atrativa e dinâmica. Identificámos rapidamente os diversos problemas, que passamos a enumerar:

- As representações geográficas existentes no *website* (Figura 7) não dispunham de uma componente dinâmica básica, como o *Zoom*. Desta forma, não era possível fazer qualquer tipo de aproximação aos pontos de interesse e perceber aspetos básicos como, a sua envolvente, a elevação de um ponto em relação a outro ou a localização exata do ponto.
- Outro problema detetado na representação dos pontos de interesse foi a impossibilidade desta representação realizar a conversão de mapa temático para imagem de satélite. Esta situação constitui um entrave significativo para o utilizador que procura esta informação, uma vez que este pode querer saber se os pontos à sua volta se situam, por exemplo, na natureza e se sim, qual o tipo de vegetação envolvente, ou, pelo contrário se são circundados por uma zona habitacional ou dotada de certas infraestruturas, como por exemplo estradas.
- Uma outra falha na representação ocorria na circunstância do mapa ter muitos pontos de interesse e deste não disponibilizar uma barra de pesquisa para realizar uma pesquisa mais eficiente. Nesta situação, a pesquisa pela informação torna-se muito demorada e pouco fluída, particularmente em casos em que o utilizador já tem conhecimento de que pontos de interesse quer visitar e utiliza o mapa somente para saber a localização dos mesmos no município.
- Outro problema do mapa prendia-se com a falta de responsividade do mesmo, ou seja, este não se adaptava às mudanças na janela do *browser*. Sucedia que quando colocávamos a janela menor, o mapa simplesmente desaparecia, consequentemente a experiência tornava-se inexistente quando redimensionávamos a nossa janela do *browser*. Sumariamente, este problema está conectado com a falta de *Zoom In* e *Zoom out*, pois se não era possível variar o zoom, o mapa quando ficava pequeno a sua informação ficava ilegível o que levou a que o desenvolvedor do *website* retirasse este conteúdo quando este ficava demasiado pequeno.

- Outro problema detetado recaía sobre a inexistência de uma escala no mapa, devido a esta falha os utilizadores não conseguiam ter noção da distância entre pontos. Sem embargo do mapa embora ter uma legenda, esta dizia essencialmente o mesmo em todos os mapas do *website* e continha informação não requisitada pelo utilizador naquela fonte. Acrescentamos que todas as representações de pontos no mapa consistiam somente em pontos coloridos, sem qualquer alusão ao que era procurado.

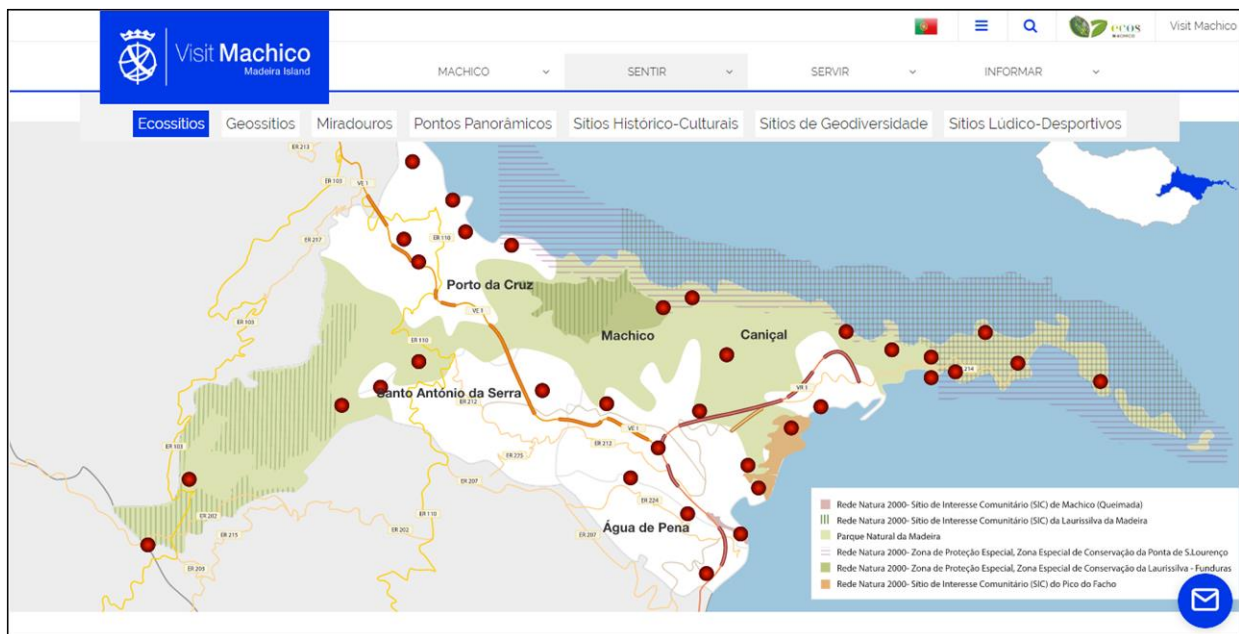


Figura 7: Mapa do *website* Visit Machico. Fonte: <https://www.visitmachico.com/pt/sentir/ecossitios/encontre-o-seu-ecossitio> Acesso em: 17 junho 2019.

A representação geográfica que está implementada no *website* (Figura 7) não é muito mais que uma imagem dos locais de interesse. Por este motivo, a experiência no *website* torna-se muito pobre. No entanto, nem tudo são pontos negativos. Verificámos que esta representação tem alguns pontos positivos, como um pequeno *popup* que abre aquando da realização de um *mouseover* dos pontos georreferenciados, e ainda clicando nos mesmo pontos estes reencaminham-nos para o sítio do *website* onde consta informação acerca deste ponto de interesse. Mais concretamente, estes pontos estão dotados de âncoras que conferem alguma dinâmica à representação geográfica.

3.2. Análise de requisitos

A análise de requisitos tem por fim definir a estrutura e o comportamento de um software que esteja a ser desenvolvido. Cada requisito é uma premissa que possibilita que

o software possa cumprir os seus objetivos básicos, permitindo ao cliente resolver os problemas relacionados.

Os requisitos subdividem-se em requisitos funcionais e não funcionais. Os requisitos funcionais estão relacionados com as funcionalidades que o cliente deseja que a aplicação disponha, e os não funcionais com as qualidades do software.

Para definir estes mesmos requisitos foi necessário articular com a equipa do *website* VisitMachico de modo a perceber quais eram os objetivos do *website* com a aplicação e ainda que tipo de dados a CMM detinha.

Nesta fase também decidimos que não iríamos fazer recolha de novos dados, uma vez que não acrescentaria grande interesse ao projeto e atendendo também ao curto espaço de tempo que possuíamos para concluir o webSIG.

No que toca às camadas da aplicação estas serão três, camada de dados, camada aplicacional e a camada cliente (Figura 8).

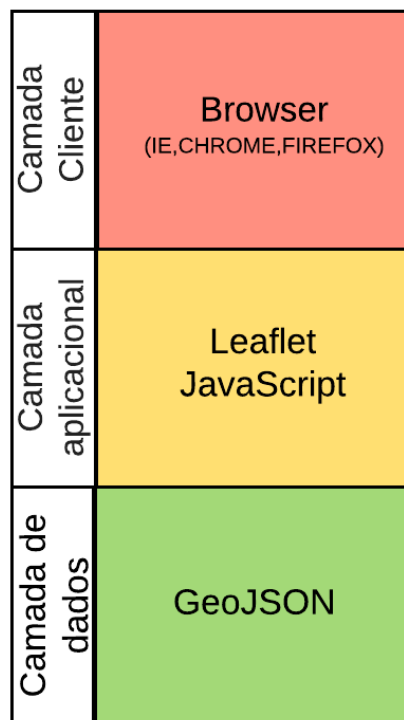


Figura 8: Camada de aplicação.

3.2.1. Quadro dos requisitos funcionais

Os requisitos funcionais que foram adotados no desenvolvimento do projeto encontram-se resumidos no Quadro 1.

Quadro 1: Requisitos funcionais

| | |
|-------|--|
| RF.01 | O mapa deve conter uma organização que auxilie os utilizadores a ter uma visão geral das funções existentes. |
| RF.02 | A aplicação deve ser dotada de imagens alusivas aos pontos de interesse de forma intuitiva para o utilizador. |
| RF.03 | O webSIG deve conter todo o mapa da Madeira e não só do Município de Machico. |
| RF.04 | A navegação deve ser simples e ter uma estrutura de fácil interpretação. |
| RF.05 | A aplicação deve disponibilizar uma ferramenta de “arrastar o mapa” por forma a poder visualizar informação espacial de forma amigável para que o resultado da vista se apresente de acordo com as necessidades do utilizador. |
| RF.06 | A solução deve conter um botão de Zoom In e Zoom Out, que também deve ser passível de operacionalizar através do scroll do rato. |
| RF.07 | Os dados têm de ser provenientes de um ambiente SIG, ou seja, estes têm de estar originalmente no formato <i>ShapeFile</i> . |
| RF.08 | A plataforma deve conseguir interpretar dados em GeoJSON. |
| RF.09 | Em casos que seja pertinente uma barra de pesquisa a solução deve apresentar uma para melhor navegação. |
| RF.10 | A solução tem de conter uma possibilidade de mudar de tipo de mapa base, passando de temático para satélite. |
| RF.11 | Cada representação deve incluir uma legenda e uma escala. |
| RF.12 | A escala deve variar consoante a janela de visualização. |
| RF.13 | A solução tem de incluir os equipamentos culturais, património, pontos de interesse, ecossítios, miradouros, praias e outros equipamentos de relevo turístico. |

3.2.2. Quadro dos requisitos não funcionais

Os requisitos não funcionais que foram adotados no desenvolvimento do projeto encontram-se resumidos no Quadro 2.

Quadro 2: Requisitos não funcionais

| | |
|--------|---|
| RNF.01 | A aplicação deve ser construída com recurso à biblioteca Leaflet. |
| RNF.02 | A solução tem de utilizar a linguagem JavaScript. |
| RNF.03 | A estilização da aplicação deve ser realizada com CSS. |
| RNF.04 | Evitar a construção de páginas "sem saída". |
| RNF.05 | A solução proposta deve estar acessível a qualquer hora e local. |
| RNF.06 | Conceber a aplicação de modo a que possa ser consultada em diferentes browsers. |
| RNF.07 | Cumprimento da denominada “regra dos três cliques”, ou seja, disponibilizar ao utilizador a funcionalidade ou conteúdo que pretende, no máximo em três cliques de rato (Zeldman, 2001). |
| RNF.08 | As janelas <i>popup</i> devem aparecer à esquerda do ecrã e deve conter uma imagem e uma descrição. |
| RNF.09 | Utilizar elementos gráficos para auxiliar o utilizador na sua seleção. |
| RNF.10 | Utilizar sempre o mesmo tipo de fonte em toda a solução. |

3.3. Desenho da aplicação

O desenho da aplicação é um processo que incorpora todas as fases anteriores. Após ler a bibliografia, entender o estado da arte no que diz respeito aos webSIGs, perceber os problemas e as mais valias da antiga aplicação e de compreender qual é o rumo que a equipa do Visit Machico quer para aplicação, encontro-me então capacitado para desenhar a futura aplicação do *website*.

A aplicação terá uma utilização simplificada, uma vez que os turistas pretendem obter rapidamente a informação, sem ser necessário recorrer a muitas pesquisas, tendo sempre em conta a "regra dos três cliques" do rato (Zeldman, 2001). Pretende-se que a aplicação seja visualmente atraente, facilmente interpretada e dinâmica, de modo a que esta pesquisa de informação não se torne uma tarefa entediante. Tendo em consideração os fatores elencados foi desenhado o seguinte esboço (Figura 9) para servir de auxílio à construção da aplicação.

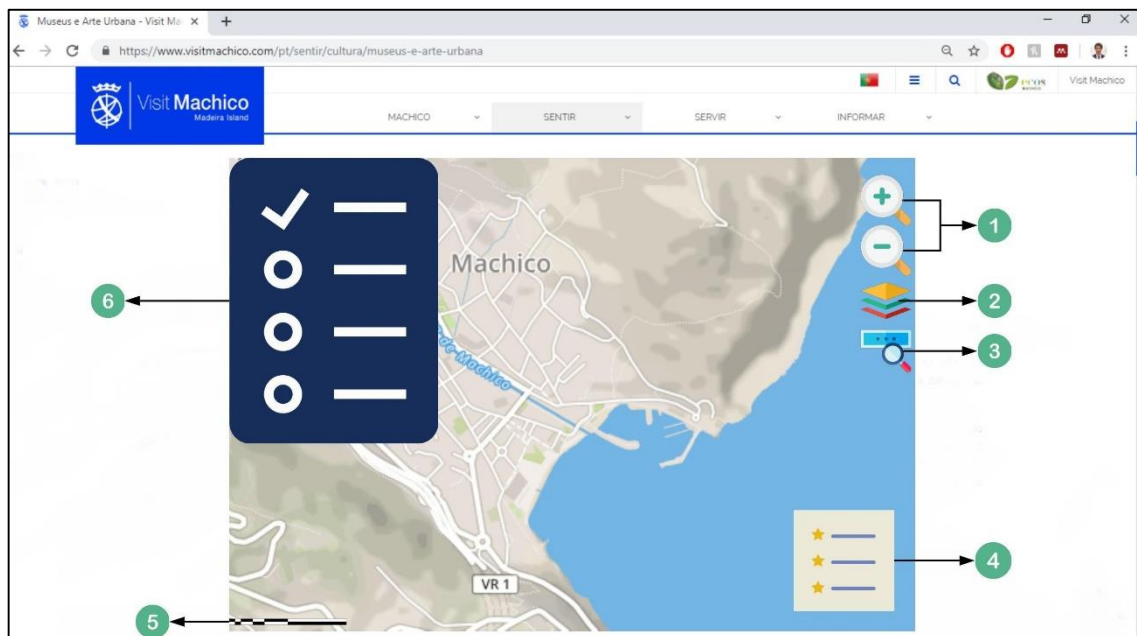


Figura 9: Esboço da aplicação.

- 1 **Zoom:** O *Zoom in* e *Zoom out* comanda a escala de visualização do mapa, é uma ferramenta indispensável para qualquer mapa interativo. O Zoom pode ser alterado clicando nos ícones de Zoom in e Zoom out, e também através do *scroll* no rato.
- 2 **Mapa de Base:** Esta ferramenta altera o mapa de base, podendo o mesmo passar de satélite para temático.
- 3 **Barra de pesquisa:** Através desta barra de pesquisa podemos fazer qualquer pesquisa dos pontos de interesse por palavra chave. A barra de pesquisa vem facilitar a nossa pesquisa quando estamos à procura de um ponto de interesse cujo nome já conhecemos. Após clicar no ponto de interesse procurado na barra, o mapa automaticamente fará um *close-up* ao ponto.

- 4 **Legenda:** No canto inferior direito estará uma legenda onde aparecerá todos os pontos de interesse que estarão representados no mapa, assim como os seus ícones identificativos.
- 5 **Escala:** No canto inferior esquerdo encontrar-se-á uma escala gráfica. Esta irá ajustar-se às diferentes alterações no Zoom.
- 6 **Descrição dos pontos de interesse:** No canto esquerdo do *browser* irá aparecer uma janela após clicarmos nos ícones/pontos de interesse do mapa. Esta janela irá apresentar diversas informações sobre o ponto de interesse selecionado.

Capítulo 4 – Camada de dados

4.1. Recolha e origem dos dados

Os dados utilizados neste projeto, provieram da base de dados do *website* Visit Machico. Embora estes necessitassem de algumas melhorias, principalmente ao nível da organização e da estrutura dos mesmos, não foi necessário ir ao campo recolher mais dados, já que os que o *website* detinha eram suficientes para abordar os tópicos necessários à conclusão do projeto.

Após analisar os dados e ter compreendido a dinâmica do *website*, optou-se por utilizar o separador “sentir” do *website* Visit Machico e esmiuçar toda a informação desse separador. O resultado foi organizado em cinco mapas com pontos de interesse para o turista.

Os cinco mapas abordarão cinco grandes temas, que serão: Museus e Arte Urbana, Património municipal, Miradouros, Mar e Ecosistemas. Sendo que o mapa dos museus e arte urbana, Património municipal e Mar se desdobram em diferentes pontos de interesse.

O mapa dos museus e arte urbana será composto por museus, solares e monumentos. O mapa do Património municipal albergará igrejas, capelas e o principal património municipal. O mapa do Mar será formado pelas praias e pelas marinas. Por último, as representações dos Miradouros e dos Ecosistemas serão exclusivamente dedicadas a estes temas.

Cada mapa terá campos diferentes no momento da abertura do toggle, a Figura 10 mostra como os campos que serão disponibilizados no mapa.

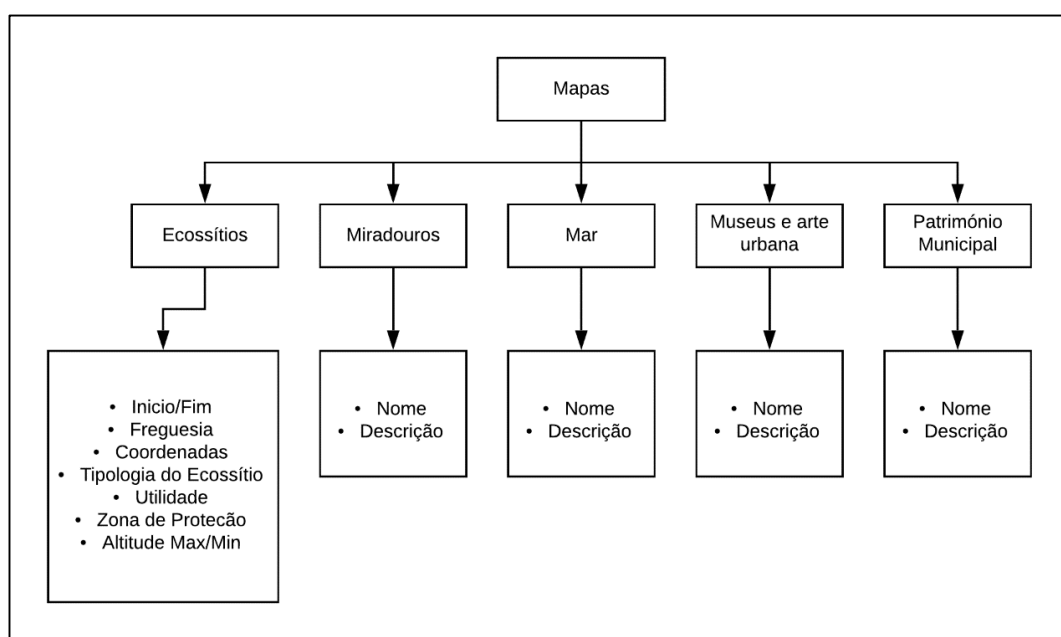


Figura 10: Esquema dos campos do toggle.

4.2. Estrutura dos dados

Embora todos os dados já estivessem recolhidos, foi necessário realizar algumas melhorias a nível da estruturação e da organização dos mesmos, sendo que todas estas alterações foram realizadas dentro do ambiente SIG, neste caso no *ArcMap* 10.6.1.

Com o avançar do projeto deparei-me com três tipos de situações nas quais os dados se encontravam. (i) Os dados estão devidamente cartografados e têm tabelas de atributos, mas não se encontram na mesma *shapefile* e a tabela de atributos está organizada de forma diferente nas diferentes *shapefiles*. (ii) Os dados têm *shapefile*, mas não têm qualquer descrição na tabela de atributos anexada a estes pontos. (iii) Os dados estão recolhidos, mas a única informação que temos sobre estes são as coordenadas, não tendo nem *shapefile* nem tabela de atributos.

Outra barreira que enfrentei ao longo do projeto foram as coordenadas, uma vez que todas se encontravam no sistema local “Datum Base SE - Porto Santo”, utilizado na RAM. Para utilizar o formato de dados GeoJSON na biblioteca de mapas Leaflet houve necessidade de converter o sistema de coordenadas dos dados originais para o sistema de coordenadas global WGS84.

4.2.1. Diferentes temas com tabelas de atributos desorganizadas

Nesta situação os dados apresentavam-se cartografados e exibiam tabela de atributos. Contudo, existem diferentes *shapefiles* para representar o mesmo ponto de interesse com tabelas de atributos organizadas de maneiras distintas.

Por exemplo, existem dois museus no concelho, só que estes encontravam-se em *shapefiles* separadas. Esta separação conduziu à realização de um *merge* às duas *shapefiles*, um procedimento simples e que resolve o problema das *shapefiles* estarem separadas.

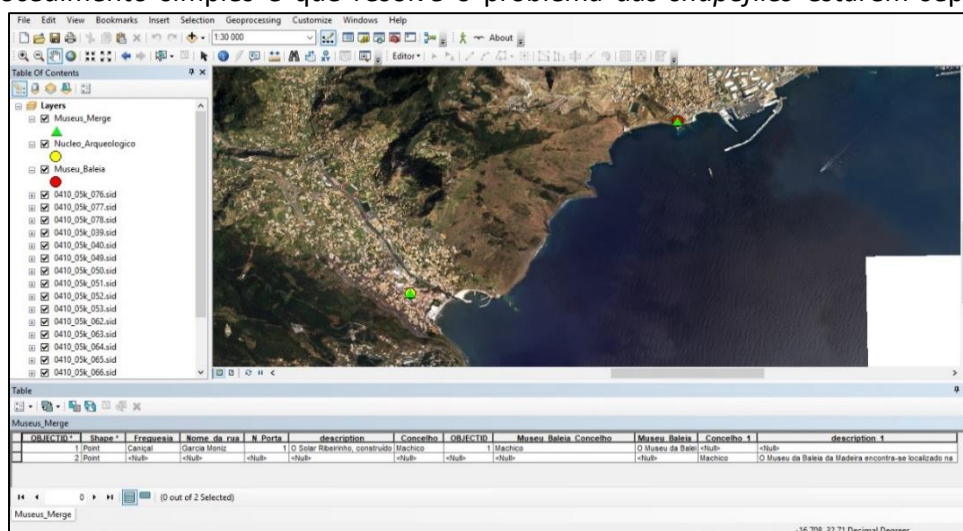


Figura 11: Diferentes *shapefiles* e com tabelas de atributos desorganizadas.

Todavia quando abri a tabela de atributos verifiquei que esta continha diversos erros (Figura 11), sendo estes: o nome das colunas não eram iguais, embora o número de registos fosse o correto existiam demasiadas colunas por registo (este erro advém do nome das colunas não ser igual e de não ser possível fazer a junção correta) e a estrutura tinha informação desnecessária para o projeto.

Tendo em conta a análise feita à tabela de atributos, decidi exportar a tabela para uma folha de Excel e trabalhar os dados a partir do Excel. Depois dos dados devidamente organizados, importamos a folha de Excel em formato .CSV para o ambiente SIG e fazemos o *join* pelo campo da chave primária, neste caso em particular o campo "OBJECTID". Neste exemplo (Figura 11), o problema podia ser resolvido no próprio ambiente SIG, visto que este exemplo apresentava poucos registos e seria facilmente manipulável a tabela de atributos. Mas para fazer tratamento de base de dados é mais indicado exportar a tabela para o Excel ou até para uma base de dados, no caso de um maior número de registos, assim conseguindo manipular melhor os dados.

4.2.2. Dados com *shapefile*, mas sem descrição na tabela de atributos e dados sem *shapefile*

Nesta segunda situação, os dados embora estivessem devidamente cartografados, a tabela de atributos não tinha qualquer descrição sobre o ponto (Figura 12).

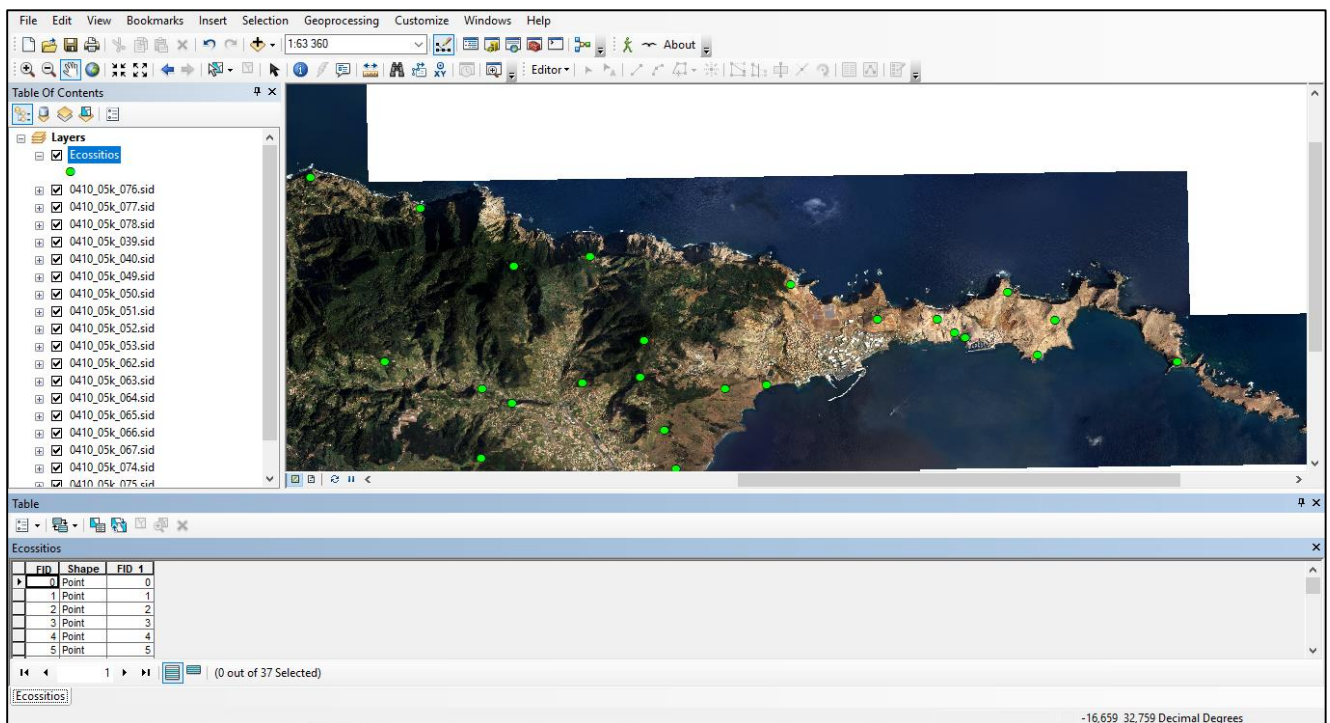


Figura 12: Dados com *shapefile*, mas sem descrição na tabela de atributos.

Colocando-se este problema a única maneira de preencher uma tabela de atributos útil para o projeto seria com conhecimento empírico e com os dados já disponíveis no *website* Visit Machico. Toda a tabela foi preenchida diretamente no ArcGIS, sendo que os dados poderiam ser colocados diretamente no GeoJSON, porém por questões de organização do projeto e para ficar com os dados todos organizados em *shapefile* foi decidido que a atualização da tabela de atributos seria feita dentro do ArcGIS.

4.2.3. Dados alfanuméricos

Os dados nesta situação eram os menos completos, pois não continham geometria. No que concerne a estes dados a única informação que detínhamos era o nome e um par de coordenadas, sendo que em alguns casos nem coordenadas existiam.

Neste caso foi necessário preencher uma tabela de atributos num Excel, onde um dos campos teria de ser as coordenadas, para posteriormente o ArcGIS reconhecer, e assim atribuir um ponto ao registo correto da tabela.

No caso em que só detínhamos o nome do ponto de interesse, a solução encontrada foi ir ao *Google Maps* e encontrar o local através de imagem de satélite e retirar assim a coordenada da localização do ponto.

4.3. Carregamento dos dados

Após todos os dados estarem organizados, devidamente georreferenciados e com a tabela de atributos que desejávamos para o projeto, seguiu-se o momento de carregar estes mesmos dados dentro do projeto. Para carregar os dados dentro do projeto a primeira etapa passou por converter todos estes dados para GeoJSON.

O formato GeoJSON advém do formato JSON, modelo de transmissão de informação no formato de texto (*string*). A grande diferença entre o formato JSON e GeoJSON é que o GeoJSON consegue suportar informação geográfica, como pontos, linhas e polígonos em forma de coordenada. Os formatos JSON/GeoJSON são muito utilizados em *web services* pois é facilmente interpretado por qualquer API no mercado e pela biblioteca que será implementada no projeto (Leaflet). No apêndice F pode ser consultado um exemplo do GeoJSON utilizado na aplicação.

Para converter os *layer* para GeoJSON no ArcGIS 10.6.1, basta ir a *conversion tools*→JSON, quando abrir a janela basta colocar o *layer* que queremos converter e ativar a *checkbox* da opção GeoJSON (Figura 13).

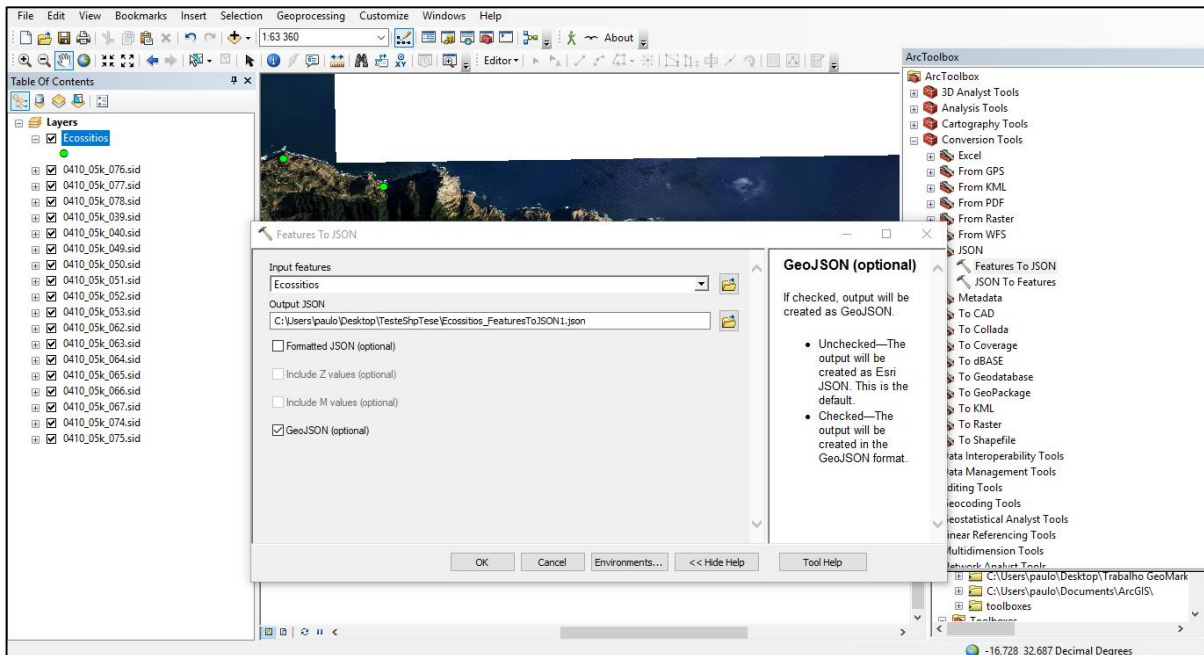


Figura 13: Conversão para GeoJSON.

Depois da obtenção de todos os dados em GeoJSON, prosseguimos com a verificação do mesmo formato, para verificar se todo o texto estava bem escrito e segundo as regras do JSON, de modo a não existirem erros ao entrar no *main script*. Para confirmar se todo o JSON estava corretamente escrito foi utilizado o *website* jsonlint.com (JSON validator).

O passo seguinte incidiu na conexão do nosso documento GeoJSON ao projeto. Para esse processo criamos um documento em JavaScript com uma variável e o código em GeoJSON. Assim, quando chamarmos essa variável no *main script*, automaticamente o *script* reconhecerá que é para representar a informação geográfica presente neste GeoJSON (Figura 14).

Outro passo para o GeoJSON funcionar corretamente consiste em verificar as coordenadas. Estas têm de se encontrar em graus decimais, sendo este o formato mais apropriado para a biblioteca Leaflet. Esta é uma das razões para a escolha do sistema de coordenadas global WGS84.

```
1 var ecossitios = {
2   "type": "FeatureCollection",
3   "name": "ecossitios",
4   "crs": {
5     "type": "name",
6     "properties": {
7       "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
8     }
9   },
10  "features": [{
11    "type": "Feature",
12    "properties": {
13      "FID": 0,
14      "Nome": "Ponta de São Lourenço",
15      "InicioFim": "Inicio/Fim: Estrada Regional 109 (Canical) / Pico da",
16      "Freguesia": "Freguesia: Canical.",
17      "Coordenadas": "Coordenadas:&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;Latitude",
18      "Distância": "Distância: 4,2 Km + 4,2 Km.",
19      "TipologiaEcossitio": "Tipologia do Ecossitio: Miradouro; Geossitio",
20      "Utilidade": "Utilidade: Caminhada; Trail.",
21      "Zonadeproteção": "Zona de proteção: RN 2000 - ZPE, SIC, ZEC; PNM.",
22      "AltitudeMaxMin": "Altitude Max/Min: 162m / 63m."
23    },
24    "Imagem": {
25      "logoLocation": "../imagens/pontaSaoLo.jpg"
26    },
27    "geometry": {
28      "type": "Point",
29      "coordinates": [-16.68108, 32.73899]
30    }
31  },
32 }
```

Figura 14: Ligação do GeoJSON com o script.

Capítulo 5 – Implementação

5.1. Linguagens e bibliotecas

Desde o início do projeto foi imperativo a aplicação ser toda concretizada com recurso a linguagens de programação.

Os servidores de aplicações de mapas fazem com que o processo de construção do webSIG seja simplificado, porém tornam o *layout* menos personalizável e mais rígido na sua estrutura. Atendendo ao facto de que um dos grandes objetivos do projeto era ser bastante personalizável e que fosse possível alterar qualquer pequeno pormenor na aplicação, recorremos a linguagens de programação cliente e a bibliotecas de apoio para concretizar a aplicação.

Como foi mencionado anteriormente no relatório, para a construção da aplicação webSIG foi necessário recorrer a diversas bibliotecas e linguagens. Nos requisitos não funcionais foram definidas à partida essas linguagens e bibliotecas.

Na aplicação irá predominar a linguagem JavaScript e a biblioteca base será o Leaflet. Embora o JavaScript e o Leaflet sejam a base, será necessário utilizar a linguagem de marcação HTML, linguagem de estilo CSS e também será utilizada a biblioteca jQuery. Embora a linguagem jQuery não esteja presente nos requisitos não funcionais, esta será necessária, pois vem simplificar o script.

5.1.1. Linguagem de marcação (HTML5)

O criador da World Wide Web e o atual diretor da W3C, órgão que padroniza a World Wide Web, foi também o criador da linguagem HTML. Formalmente apresentada na década de 90, onde as primeiras quatro grandes versões saem nesta mesma década, e só em 2014 é apresentado o HTML 5, versão esta que é utilizada atualmente (Longman, 1998; w3schools, 2019).

A grande diferença do HTML 5 para as versões anteriormente lançadas, é que esta nova versão é fortemente orientada para novas realidades tecnológicas, sendo os dispositivos móveis uma das grandes preocupações. Um exemplo claro desta orientação aos dispositivos móveis, é quando clicamos em algum elemento que necessite de campos de entrada. Ao clicar no campo e quando a aplicação reconhecer que está a ser aplicada num navegador web móvel, este automaticamente irá dispor um teclado virtual, permitindo assim que o utilizador preencha os campos através do dispositivo móvel. Desta maneira o programador não tem de se preocupar com a introdução destas funcionalidades específicas para dispositivos móveis (Valério, 2014).

O HTML é uma linguagem de marcação, que se resume a um conjunto estruturado de instruções, conhecidas como *tags*, que transmitem ao navegador como estruturar o *website*, ou seja, o navegador interpreta essas instruções e desenha a estrutura que foi solicitada (Caldeira, 2015).

Todos os documentos HTML têm uma estrutura básica, que é composta pelas seguintes *tags*: **html**, **head**, **title** e **body**. Para além destas *tags* o documento deve conter uma informação DOCTYPE na primeira linha, para que o navegador saiba qual é a versão da linguagem que estamos a utilizar.

Tag html- A *tag* html delimita todo o documento, assim todas as *tags* do documento devem estar dentro da *tag* html.

Tag head- Tudo o que for colocado na *tag* head não será visível no navegador, mas contém instruções sobre o conteúdo e comportamento, como por exemplo, scripts ou folhas de estilo (css).

Tag title- Esta *tag* tem como objetivo definir o título da página e por sua vez o nome que aparece na aba do navegador.

Tag body- O body é apresentado visualmente no navegador e deve conter todo o conteúdo da página e suas respectivas *tags*, como botões, títulos, imagens, entre outras.

5.1.2. Linguagem de estilo (CSS)

Quando o HTML foi criado, a sua principal função não era estilizar a página, mas à medida que este vai crescendo em popularidade vai surgindo a necessidade de mudar a aparência do documento. Esta necessidade fez com que incluíssem ferramentas para alterar o HTML nesta mesma linguagem. Este processo de incluir a estilização do documento dentro do HTML fez com que esta linguagem se tornasse muito complexa e difícil de entender e também trazia problemas de compatibilidade entre *browsers*, o que levou a que nesta altura, no início da década de 90 houvesse dificuldades por parte dos desenvolvedores de *websites* (Bos, 2016).

Em 1994 Håkon Wium Lie que trabalhava no CERN, onde a web estava a começar a ser usada como plataforma para publicações eletrónicas, deparou-se com o problema de estilização do HTML. Percebendo estas mesmas dificuldades Håkon Lie resolveu criar uma maneira mais simplificada de formatar a informação do HTML, foi nesse mesmo ano que Håkon Lie propôs a criação do CSS (Bos, 2016).

Em 1995 e já trabalhando com Bert Bos, que na altura trabalhava para um motor de busca chamado de Argo, apresentaram a proposta do que seria o CSS. A proposta teve tal impacto que chamou a atenção da W3C, o que fez com que a W3C criasse uma equipa para este projeto, liderada por Håkon Wium Lie e Bert Bos (Bos, 2016).

Em resultado desta colaboração surgiu logo em 1996 o CSS1, em 1998 o CSS2 a segunda versão da folha de estilo para a web. Atualmente é utilizado o CSS3 sendo compatível com todos os grandes motores de busca do mercado, dificuldade do CSS2, como o Google Chrome, Opera, Internet Explorer, Safari e Mozilla Firefox, tornando assim mais fácil a tarefa de quem programa para a web e para os próprios utilizadores.

O CSS é utilizado para estilizar, ou seja, para definir a aparência das páginas da internet. Esta estilização vai desde a cor de background, tipo de título, tipo de fonte da página, design da página entre outras funções. Para tal é criado um arquivo externo que contém as regras aplicadas, ficando assim mais fácil aplicar estilo a um *website*.

5.1.3. Linguagem JavaScript

O criador do JavaScript, Bredan Eich, foi recrutado pela Netscape Navigator Sun em 1995, onde começou a trabalhar no projeto Mocha, nome de projeto da linguagem JavaScript. Nesta mesma data, mais precisamente em maio de 1995, Bredan Eich escreveu o seu primeiro protótipo do JavaScript em apenas dez dias (Lebec, 2019).

A linguagem JavaScript teve diversos nomes antes do seu nome final. Na altura que estava a ser desenvolvida o nome desta era Mocha, posteriormente passou a chamar-se LiveScript quando foi lançada a versão beta e, finalmente na sua versão final, designou-se de JavaScript, devido às suas similaridades com a sintaxe do Java, embora estas linguagens não tenham nenhuma outra similaridade. Esta semelhança dos nomes entre estas duas linguagens atualmente ainda causa confusão em alguns utilizadores (Grillo & Fortes, 2008).

O JavaScript possibilita a criação de pequenos programas dentro do código de um HTML, sendo que estes programas são capazes de verificar formulários, processar dados, alterar valores de um elemento no HTML, criar elementos entre outras coisas, tornando assim o HTML mais dinâmico e mais rápido. Todo este processo acontece do lado do cliente, evitando a troca de informação cliente servidor que acontecia antes do JavaScript. Este passo foi importante na ocasião, pois a velocidade da internet era pouco satisfatória, assim sendo, o tempo de espera de qualquer ação passa a depender da velocidade de processamento do cliente e não da velocidade da internet (Grillo & Fortes, 2008).

O JavaScript é uma linguagem de alto nível e dinâmica, direcionada para uma programação orientada a objetos. Apesar do JavaScript ser uma linguagem muito ligada ao *front-end*, isto é, à parte do cliente, parte visível para o utilizador, esta também consegue ter valências no *back-end*, através do interpretador Node.js, criado para trabalhar no lado do servidor através da linguagem de JavaScript. Tornando assim o JavaScript numa das linguagens mais versáteis do mercado (Flanagan, 2011).

Atualmente existem interpretadores de JavaScript na grande maioria dos sites e navegadores, mas também podem ser encontrados em computadores, consolas de jogos,

tablets ou smartphones, tornando esta numa das linguagens mais omnipresentes do momento.

5.1.4. Biblioteca Leaflet

Leaflet é uma biblioteca JavaScript de código aberto que foi criada em 2011 por Vladimir Agafonkin com partes do código da API do “*Web Maps API*”, sendo esta uma biblioteca utilizada para o provedor de mapeamento CloudMade, local onde Agafonkin trabalhava em 2010. Os grandes objetivos de Agafonkin era que o Leaflet fosse simples, leve, fácil de desenvolver com recurso à mesma e principalmente que fosse responsiva, ou seja, compatível com todas as plataformas moveis e em *desktop* (CloudMade, 2011).

Desde 2011 que a biblioteca Leaflet tem sido atualizada constantemente, mas para quem trabalha com SIG uma das atualizações é particularmente interessante. A versão 0.6 de junho de 2013, para além de ter sido melhorada a API, de terem sido corrigido erros, melhorado os controles, também foi introduzida a possibilidade de introduzir o layer no formato de GeoJSON. Esta atualização é importante pois é um formato fácil de converter nos programas de SIG, o que torna mais fácil a introdução de um grande volume de dados no *script* (Leaflet, 2019).

O grande concorrente direto do Leaflet é o OpenLayers, pois ambos são uma biblioteca de JavaScript, têm o código aberto e são ferramentas de mapas cliente. A biblioteca Leaflet é mais leve que o OpenLayers e tem uma base de código mais recente o que torna este compatível com os últimos recursos do JavaScript, HTML 5 e CSS 3. Os grandes contras do Leaflet em comparação com o OpenLayers é que este possui o recurso Web Feature Service (WFS) e o suporte nativo para projeções. O Leaflet também tem concorrentes de código fechado como é o caso dos gigantes Google Maps e Bing Maps.

Em suma, o Leaflet tem grandes mais valias para quem quer trabalhar com uma biblioteca de JavaScript, pois tem uma API bem estruturada, é responsiva, usa os últimos recursos do JavaScript, HTML 5 e CSS 3, é fácil de usar, está bem documentada e para quem quiser contribuir tem um código-fonte simples.

5.1.5. Biblioteca jQuery

A biblioteca de JavaScript de código aberto jQuery começa a ser desenhada em 2005 por John Resig. Nesta ocasião Resig tinha criados inúmeros *websites* e projetos e gostava de ter algumas ferramentas criadas por si para tornar o seu processo de desenvolvimento mais fácil, uma vez que este achava que as ferramentas de desenvolvedor do JavaScript da altura poderiam ser melhores e, sobretudo, ficava frustrado com a dificuldade em escrever JavaScript que funcionasse em vários navegadores (Khan Academy, 2017).

Devido a estas frustrações como desenvolvedor, John Resig decide criar a sua própria biblioteca de JavaScript de código aberto chamada de jQuery. Apresenta a ideia em uma palestra no BarCamp, em janeiro de 2006, para um pequeno grupo de desenvolvedores web, e faz uma publicação no seu blog a anunciar o lançamento da biblioteca. Esta foi muito bem recebida pelos desenvolvedores web e tornou-se em uma das mais populares bibliotecas de JavaScript, estando presente no código de grande parte dos grandes sites mundiais (Resig, 2016).

O slogan da jQuery diz muito sobre a ideia conceptual da mesma, “write less, do more”. Assim sendo, a biblioteca jQuery tem como objetivo facilitar o trabalho em JavaScript simplificando os *scripts client-side* que interagem com o HTML. Este faz com que o programador escreva menos linhas e reutilize o código. O jQuery também veio resolver o problema de incompatibilidade entre os navegadores (problema muito presente na época), trabalha com AJAX e é uma biblioteca muito leve (Tutorials Point, 2015; York, 2015).

5.2. Desenvolvimento do *script*

Este ponto do relatório tem como objetivo esmiuçar o *script*, demonstrar como este foi realizado e apontar os principais problemas e as principais vantagens de abordar um webSIG através destas linguagens de programação e da biblioteca Leaflet. Os scripts das aplicações dos mapas de ecossítios, mapa mar, mapa miradouros, mapa museus e arte urbana e o mapa de património municipal podem ser consultados nos apêndices A,B,C,D e E.

Para explicar o script vou usar um dos cinco mapas que foram realizados para este projeto, neste caso vou usar o mapa dos ecossítios. Vou explicar o script de cima para baixo pela estrutura do HTML, esta ordem nada tem a ver com a ordem de construção do mesmo.

5.2.1 <Head>

<**title**>- O title é o título que queremos que apareça na aba, quando esta é aberta.

<**meta**>- O meta tem como função indicar a codificação dos caracteres da página, neste caso e na grande maioria dos casos usamos o Unicode UTF-8, pois esta codificação tem todos os caracteres necessários na língua Portuguesa.

<**link**>- Estes links fazem a ligação entre este documento e um recurso exterior. Estes links podem ser internos (âncoras), locais e globais. No caso deste mapa temos vários links locais e globais. Podemos observar a ligação que temos por exemplo, ao *stylesheet* do CSS. Apesar de podermos colocar o CSS neste mesmo *script*, por uma questão de organização e de limpeza do código, preferi colocar à parte, e chamar estas folhas de estilo (CSS) através

de links. Também podemos ver um `<link>` que vai buscar uma fonte do google, pois na descrição dos pontos de interesse decidi usar outro tipo de letra que só estava disponível no google *fonts*.

`<script>`- Esta tag é usada para incluir ou referenciar um script executável. No mapa dos ecossítios temos vários *scripts* externos que queremos que sejam executados no nosso mapa, sendo na sua maioria as bibliotecas usadas. Como podemos ver no *script* todas as bibliotecas que são chamadas são acompanhadas com a versão que está a ser chamada.

5.2.2 `<Body>`

`<div>`- Nesta div temos o toggle, isto é, a área que vem do “exterior” do mapa, quando clicamos num ponto de interesse. Nesta parte do *script* temos toda a estrutura do toggle, como os tópicos que estão no toggle, o botão para fechar o mesmo quando desejarmos e a imagem que irá aparecer. Nesta situação em caso de falha no carregamento da imagem coloquei a mensagem automática “Problemas a carregar a imagem”.

No body ainda temos outra `<div>` que corresponde ao identificador do mapa.

5.2.2.1 `<Script>`

Na tag `<script>` colocamos as alterações dinâmicas que queremos que o JavaScript realize na nossa página, como por exemplo formulários, imagens ou botões que tenham qualquer ação (como chamar toggle ou popups). É nesta tag que vamos definir e chamar onde o mapa vai abrir, os mapas de base que vamos usar, os ícones que vão ser colocados no mapa, as imagens que terão esses ícones, dizer onde o toggle vai buscar a informação para se preencher automaticamente (que neste caso é a um GeoJSON), o controlador do zoom, a escala, a barra de procura e a legenda. Basicamente todo o que tenha qualquer dinâmica foi colocada nesta tag.

No mapa, temos dentro da tag `<script>` duas coisas, variáveis ou funções. O *var* vai definir variáveis para um problema matemático em contexto de programação e o *function* é executada quando é chamada e é desenhada para fazer determinada tarefa.

Um exemplo de *var* no código do mapa, é quando criamos a variável para todo o mapa, que se chamou “*var mapaCmm*”, ou seja, quando passar a chamar “*mapaCmm*” vou me estar a referir à variável que foi definida anteriormente. Esta variável tem várias propriedades, nomeadamente como “*seetView*”, em que as coordenadas, e a distância que queremos que o mapa seja chamado ou até o zoom máximo ou mínimo que queremos que o mapa faça (neste caso escolhi um zoom mínimo de 10 para não sair do território da RAM).

Um exemplo no mapa de uma função trata-se da situação na qual queremos mudar de imagem do ponto de interesse quando o rato passa por cima deste. As linhas de código que têm a função que faz com que isso aconteça são as seguintes “`layer.on("mouseover", function (e) {{e.target.setlcon(ecossitioon)}});`” e “`layer.on("mouseout", function (e) {{e.target.setlcon(ecossitiooff)}});`”

Outra particularidade que podemos ver dentro da tag `<script>` são caracteres que fazem referência a uma biblioteca, ou seja, caracteres que chamam a ação dessa biblioteca. Como no caso da biblioteca Leaflet é chamada por `L`. e no caso do jQuery é chamado através do caracter `$`.

5.2.3. Mapas de base

Para este projeto optamos por dois mapas de base, um com objetivo de ser mais temático e outro teria de ser imagens de satélite.

O *website* <https://leaflet-extras.github.io/leaflet-providers/preview/>, facilitou a tarefa de escolha dos mapas de base, este *website* permite pré-visualizar diversos mapas de base e possui a variável pronta a incorporar no código para facilitar o modo como o mapa é inserido no *website*.

O mapa de base temático pertence à OpenStreetMap Sweden, e tem por nome Hydda_Full. Optei por este mapa de base por ser “limpo” visualmente, significa isto que este mapa não tem muito ruído ou pormenor, contudo possui tudo o que um utilizador necessita para entender o espaço que vai visitar. O mapa com imagens de satélite é fornecido pela ESRI, e tem por intuito que os utilizadores percebam a envolvente e a realidade do espaço em que se encontram os pontos de interesse.

5.3. Funcionalidades da aplicação webSIG

A principal funcionalidade da aplicação é que os utilizadores consigam entender não só a disposição espacial dos diversos pontos de interesse no município de Machico, como também as principais informações sobre estes, sem perderem muito tempo em pesquisas.

Tendo em mente esta funcionalidade, a aplicação tem um *design* bastante minimalista, de modo a que o seu entendimento seja de fácil compreensão, e para que seja aplicada a “regra dos três cliques” no rato, um dos objetivos dos requisitos não funcionais (Zeldman, 2001).

Ao abrir a aplicação (Figura 16) temos diversas ferramentas, como botões de zoom, possibilidade de alteração de mapa de base, uma barra de pesquisa, legenda do mapa, escala e um *toggle* que abre ao clicar em qualquer ponto de interesse no mapa. Este *toggle* é acompanhado com informação sobre o mesmo ponto de interesse e com uma imagem do mesmo (Figura 15). No caso de já se conhecer o nome do ponto de interesse e se querer encontrar, existe a possibilidade de fazer uma pesquisa e quando se clica nesse ponto este faz um zoom automático (Figura 17, versão móvel). Todas estas ferramentas são passíveis de ser usadas em dispositivos móveis, sendo esta uma das grandes mais valias de usar o Leaflet.



Figura 15: Obtenção de informação relativa à Estátua de Tristão Vaz Teixeira.

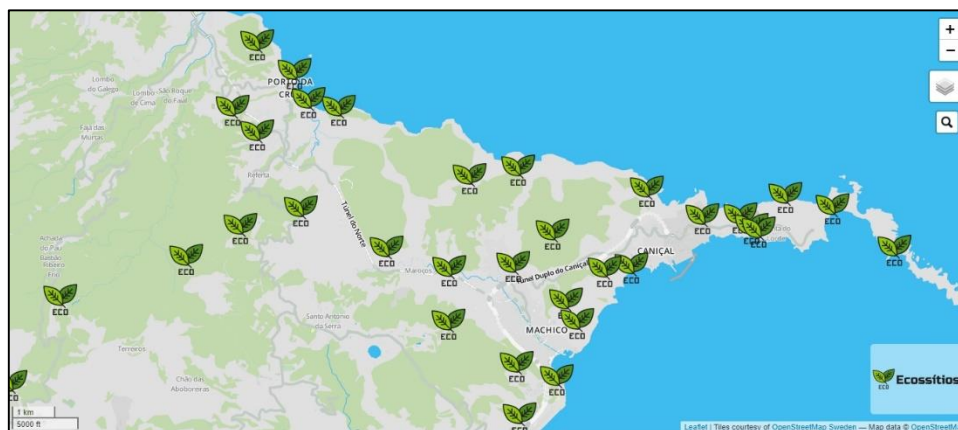


Figura 16: Layout de abertura do mapa Ecosítios.

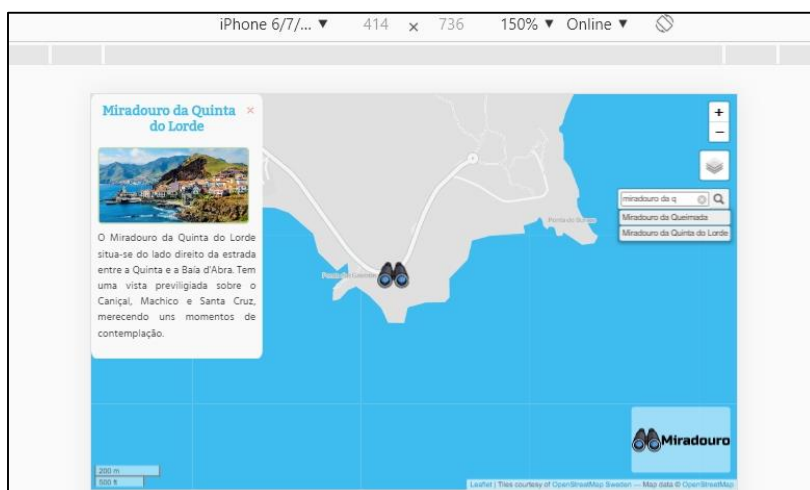


Figura 17: Zoom in após pesquisa do Miradouro da Quinta do Lorde, na versão móvel.

5.3.1. Descrição dos vários mapas

No projeto foram construídos cinco mapas com diferentes temas, sendo estes o mar, miradouros, património municipal, ecossítios e museus e arte urbana. Todos estes temas estavam presentes no *website* VisitMachico e foram criados para simplificar a pesquisa dos pontos de interesse pelos utilizadores.

No mapa com o tema mar (Figura 18) podemos encontrar diversos pontos de interesse separados em dois subtemas: marinas e praias. Nestes mapas estão representadas as marinas do porto de recreio de Machico e a marina da Quinta do Lorde. Podemos ainda encontrar praias, como a praia de areia do centro de Machico ou a Prainha.

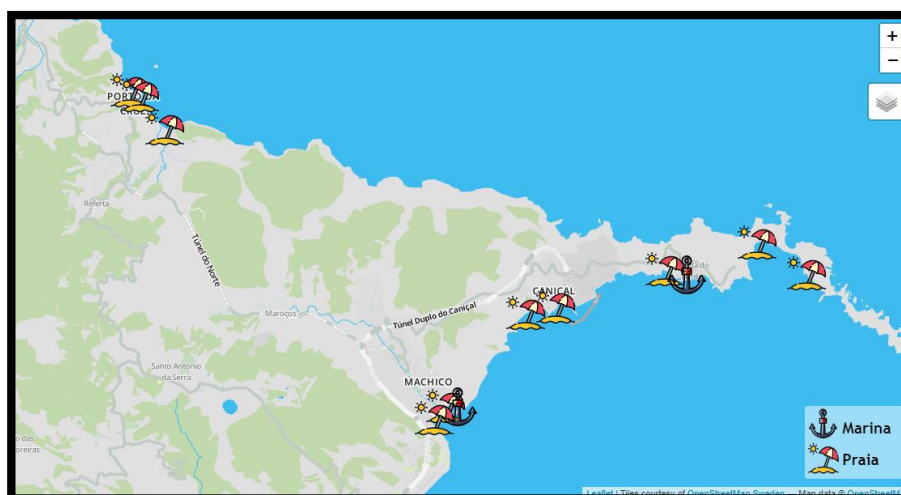


Figura 18: Mapa com o tema Mar.

No mapa dos miradouros (Figura 19) temos a descrição e a imagem de dezanove miradouros dispersos por todo o município, sendo que todos eles possuem uma vista panorâmica e têm maioritariamente o azul do oceano Atlântico como fundo. Neste mapa encontram-se miradouros como o do Pico do Facho, miradouro Francisco Alvares de Nóbrega ou o miradouro da Portela. Como este mapa tem muitos pontos de interesse coloquei a barra de pesquisa para facilitar a sua utilização.

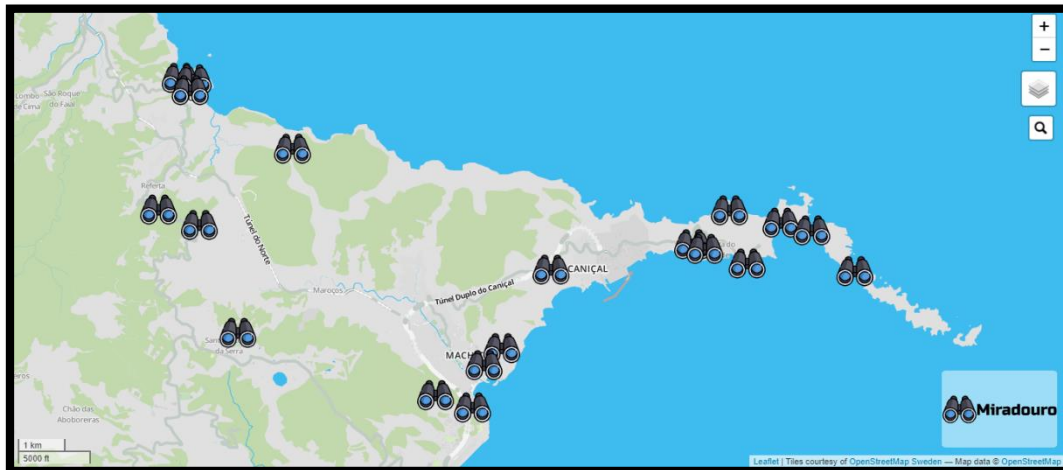


Figura 19: Mapa com o tema Miradouros.

O mapa sobre o património municipal (Figura 20) divide-se em três subtemas, como património municipal, igrejas e capelas. Entre os pontos de interesse estão o forte de Nossa Senhora do Amparo, a Igreja de Nossa Senhora da Conceição ou a Capela de São Roque.

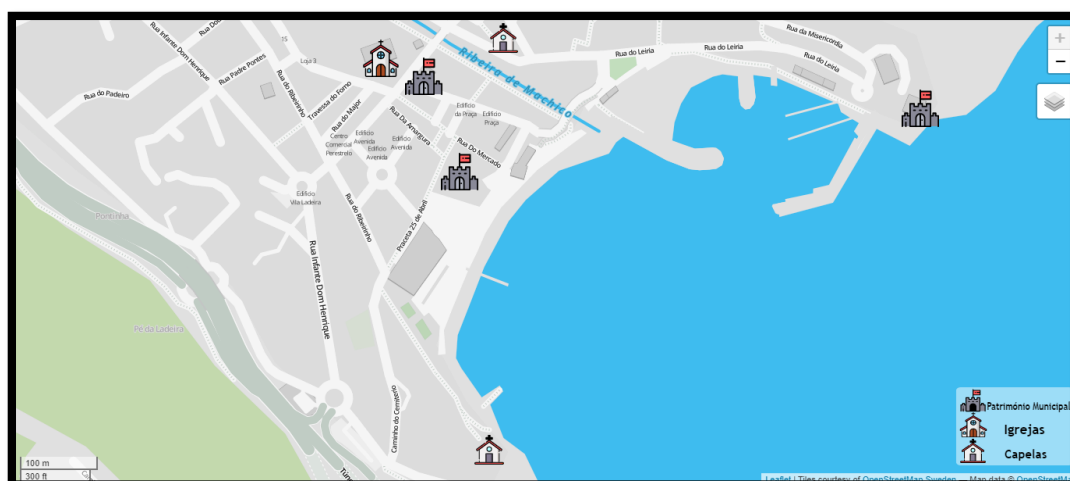


Figura 20: Mapa com o tema Património municipal.

No mapa ecossítios (Figura 21), podem encontrar-se mais de trinta ecossítios, que são pontos onde, em comparação com a sua envolvente, se encontram elementos do

patrimônio geológico que constituem uma ocorrência de reconhecido valor científico. É de ressaltar que essa importância não se restringe apenas ao fator geológico, mas também ao didático, cultural e estético. Como este mapa tem muitos pontos de interesse podemos verificar que detém uma barra de pesquisa.

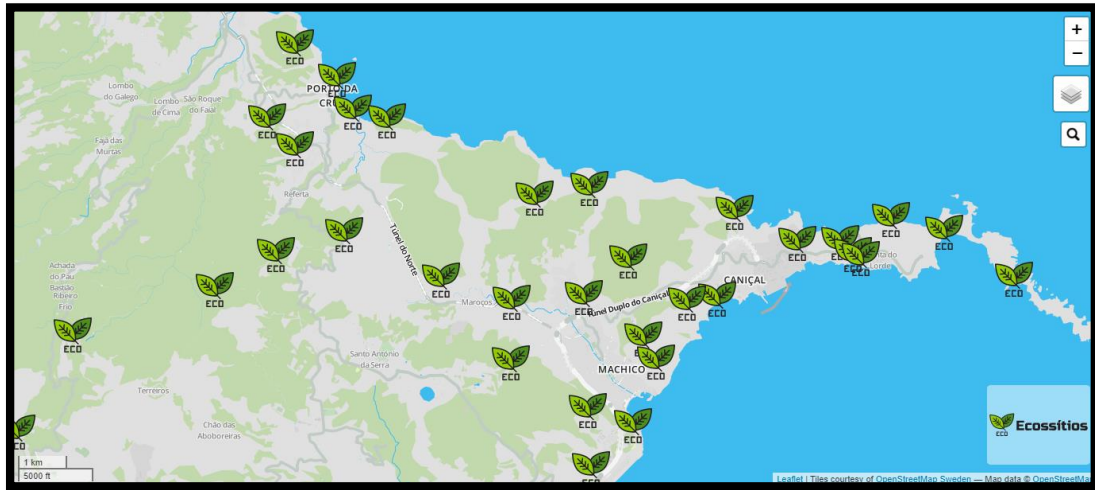


Figura 21: Mapa com o tema Ecossítios.

No mapa de museus e arte urbana (Figura 22), têm-se três subtemas, são estes museus, solares e monumentos. Neste mapa encontramos o Museu da baleia que fica na pequena vila piscatória do caniçal, o núcleo arqueológico Solar do Ribeirinho ou monumentos como a estátua de Tristão Vaz Teixeira entre outros pontos de interesse culturais.

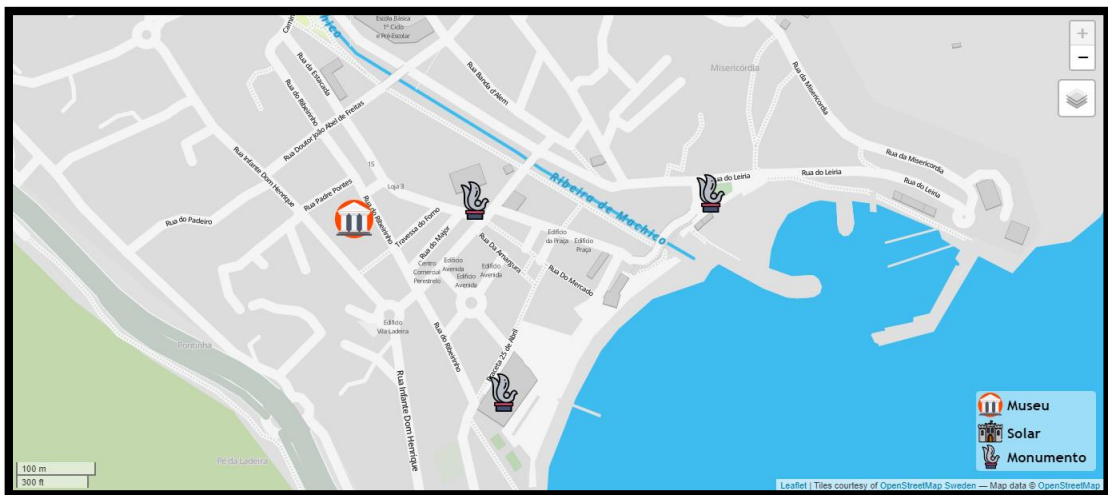


Figura 22: Mapa com o tema Museus e Arte.

5.4. Casos de estudo

Neste capítulo iremos fazer alguns testes de performance de maneira a simular uma pesquisa dentro da aplicação. Desta maneira verificamos se está tudo correto com a aplicação.

Serão realizados três casos de estudo:

1. Neste primeiro caso o utilizador procura por uma praia. Quer perceber informações sobre esta, a sua localização e a sua envolvente.
2. No segundo caso o utilizador sabe o que quer ver e só gostaria de ter mais informação sobre o ponto de interesse.
3. No terceiro caso o utilizador está na rua e gostaria de através do seu telemóvel ter acesso a informação do monumento que está a sua frente.

Caso de estudo n.º 1 - Como podemos verificar na Figura 23 ao passar o rato por um ponto de interesse este muda de cor, o que torna a experiência agradável visualmente. Quando clicamos no ponto de interesse aparece um toggle com a informação, neste caso da praia procurada (Figura 23).

Para perceber a envolvente basta mudar o mapa para a visão de satélite, ferramenta esta que se encontra à direita. Também podemos fazer um zoom in para ter mais pormenor da envolvente (Figura 24).

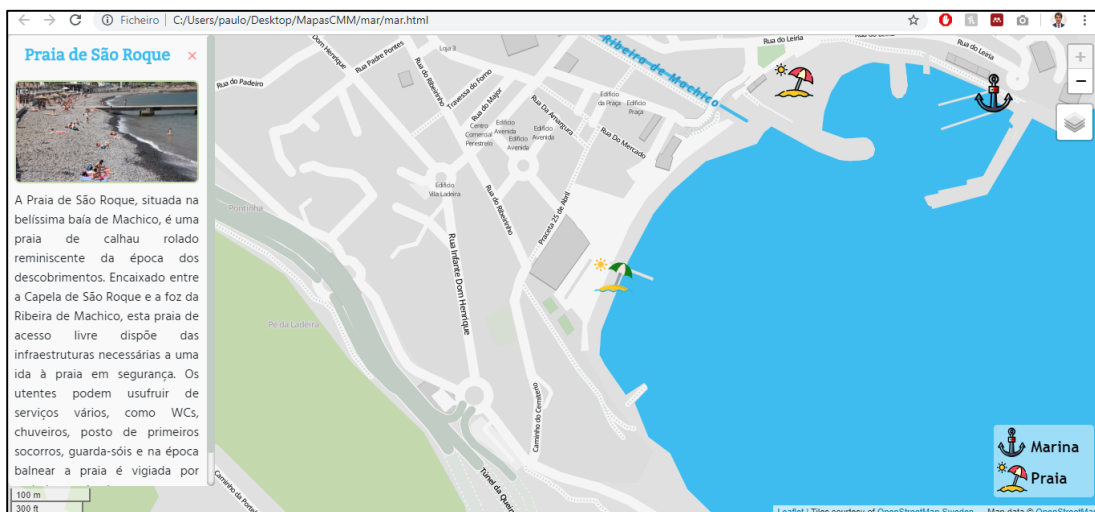


Figura 23: Ícone com mudança de cor, após a passagem do rato.

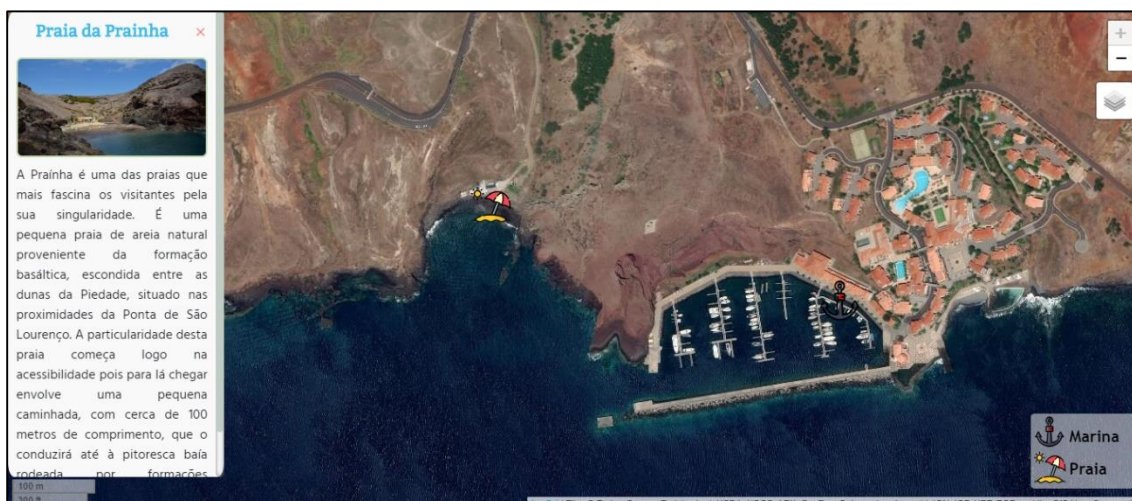


Figura 24:Toggle ativo e imagem de satélite.

Caso de estudo n.º 2 - O utilizador sabe a informação que quer retirar e o ponto de interesse que procura. Para tal o utilizador vai à zona do site chamada de ecossítios, encontra o mapa e faz uma pesquisa na barra de pesquisa por Penha D'Águia na barra de pesquisa localizada à direita (Figura 25). Ao colocar a letra "P", aparece todos os pontos de interesse do mapa em questão com a letra procurada, e se for necessário tem uma *sidebar* para ajudar na procura do ponto de interesse pela pesquisa (Figura 25). Quando o utilizador clica no ponto de interesse na barra de pesquisa, automaticamente a aplicação faz um zoom ao ponto.

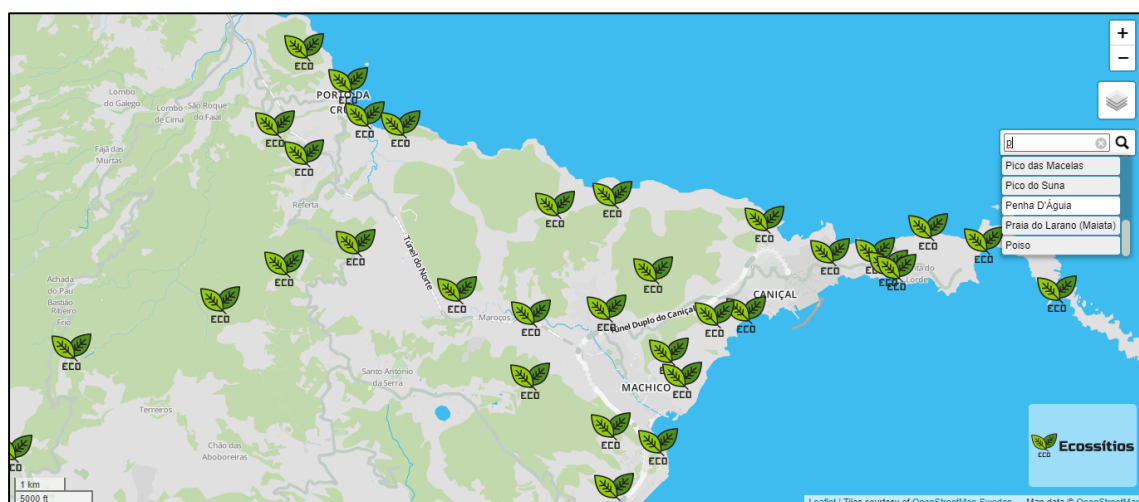


Figura 25: Mapa Ecosítios com destaque da barra de pesquisa.

Para retirar informação do ponto de interesse é só clicar no ícone, o que faz com que apareça um toggle com informação, como por exemplo a freguesia, a altitude, a tipologia do ecossítio ou as coordenadas.

Caso o utilizador queira perceber a quantidade de ecossítios que rodeia este é só fazer zoom out. O zoom out máximo é o arquipélago da madeira, como podemos ver na Figura 26.

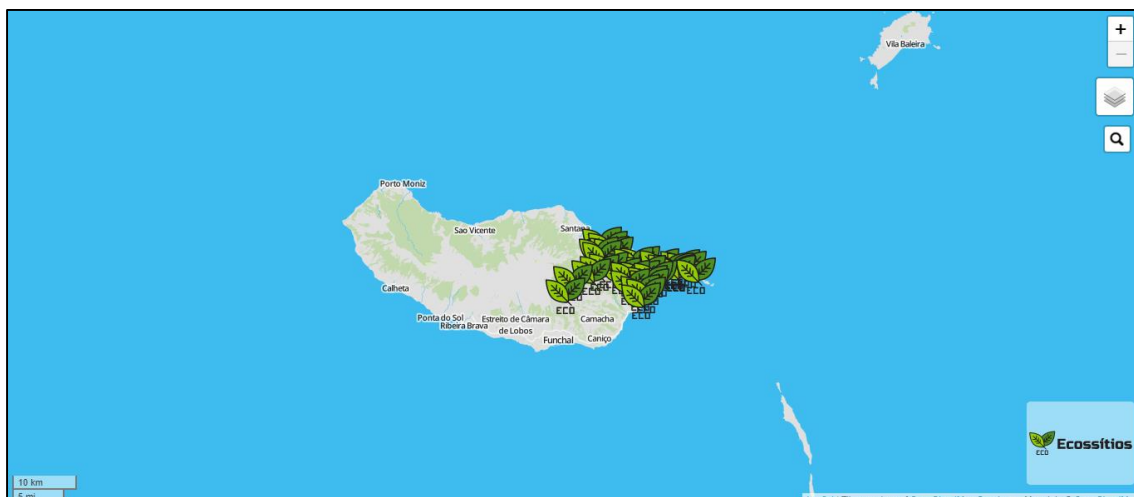


Figura 26: Zoom out máximo do mapa EcoSítios.

Caso de estudo n.º 3 - Um visitante no município de Machico ao encontrar um ponto de interesse procura no site VisitMachico o WebSIG e utiliza-o num dispositivo móvel para procurar mais informação relativa àquele ponto de interesse. Toda a aplicação funciona de igual maneira na versão móvel, ou seja, é completamente responsiva (Figura 27).

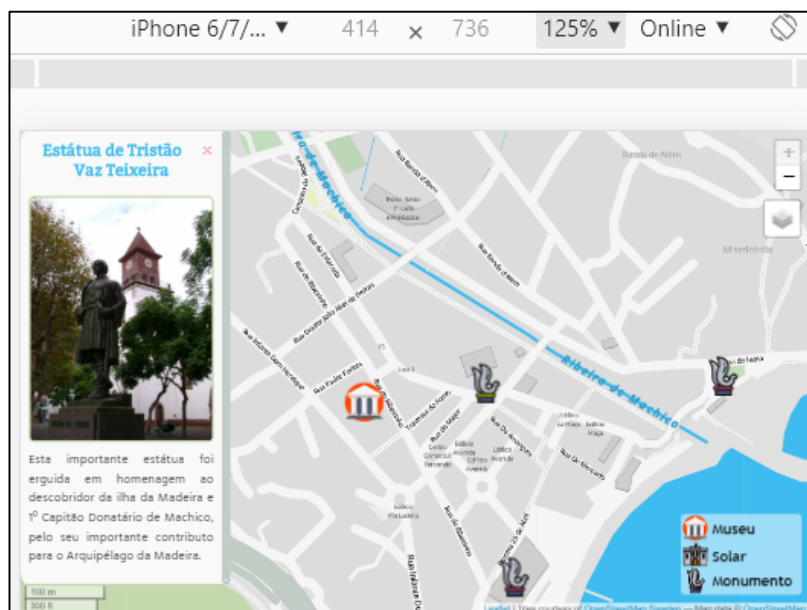


Figura 27: Versão móvel do mapa Museus e Arte

Capítulo 6 – Resultados e conclusões

6.1. Alteração da aplicação a posteriori

A criação desta aplicação teve sempre como objetivo que esta fosse fácil de alterar a posteriori, atendendo ao facto das futuras alterações necessárias serem realizadas por um técnico da câmara municipal com poucos conhecimentos de linguagens de programação.

Apesar da aplicação ser objetivamente fácil de atualizar, para fazer este processo é necessário alguns conhecimentos de JavaScript e de GeoJSON. A pessoa que realizar essa atualização necessita de ter o mínimo conhecimento de leitura de um script de JavaScript e de GeoJSON, pois para realizar qualquer alteração nos dados é necessário recorrer a GeoJSON e dar entrada no script dessa mesma alteração. Nas imagens presentes em baixo encontram-se algumas alterações rápidas possíveis de realizar sem fazer alterações complexas no código.

Para alterar o zoom mínimo e o zoom máximo possível de realizar no mapa, basta alterar os números na linha de código `minZoom` e `maxZoom` (Figura 28). Neste caso o zoom mínimo só vai até 10, uma vez que eu queria restringir o zoom mínimo ao arquipélago da Madeira. No `setView` controla-se a janela de abertura (Figura 28) do momento em que a aplicação é inicializada, nesta situação eu coloquei as coordenadas de Machico e o zoom 13, que fica só com o concelho de Machico visível.

```
// Ligação do meu script ao Identificador e algumas opções do mapa (como a janela de abertura)
var mapaCmm = L.map('mapid', {
  minZoom: 10,
  maxZoom: 17, //Maximo tem que ser 17 (0-18), pois à problemas com um basemap caso seja 18
  zoomControl: false
}).setView([32.745, -16.75], 13);
```

Figura 28: Alteração do zoom e da janela de abertura

Nesta parte do script (Figura 29), pode alterar-se a dimensão dos ícones que aparecem no mapa, em `iconSize`. Pode-se também modificar o ícone, bastando para tal mudar o nome do arquivo svg e o próprio arquivo na pasta raiz do mesmo.

```
// Defenição dos ícones do mapa
var edificiooff = L.icon({
  iconUrl: 'edificiooff.svg',
  iconSize: [48, 58]
});

var edificioon = L.icon({
  iconUrl: 'edificioon.svg',
  iconSize: [48, 58]
});

var igrejaoff = L.icon({
  iconUrl: 'igrejaoff.svg',
  iconSize: [48, 58]
});

var igrejaon = L.icon({
  iconUrl: 'igrejaon.svg',
  iconSize: [48, 58]
});

var capelaoff = L.icon({
  iconUrl: 'capelaoff.svg',
  iconSize: [38, 48]
});

var capelaon = L.icon({
  iconUrl: 'capelaon.svg',
  iconSize: [38, 48]
});
```

Figura 29: Alterar ícones.

Para alterar o campo de pesquisa da barra de procura (Figura 30), deve alterar-se o `propertyName`, que no caso deste mapa é retirado no campo “Nome_” no arquivo GeoJSON, uma vez que quando se começa a digitar na barra faz a pesquisa pelo nome do ponto de interesse, contudo essa pesquisa podia ser realizada por outro campo, como por exemplo ID, data, localidade ou ano de construção.

```
// Barra de procura
var searchControl = new L.Control.Search({
  layer: miradouro,
  propertyName: 'Nome_',
  marker: false,
  moveToLocation: function (latlng, title, map) {
    var zoom = map.getBoundsZoom(miradouro.getBounds());
    map.setView(latlng, 16); // zoom de procura
  }
});
```

Figura 30: Alterar a Barra de pesquisa.

Na Figura 31 pode ver-se o GeoJSON, aqui pode alterar-se toda a informação dos dados, isto é, tudo o que aparece no toggle.

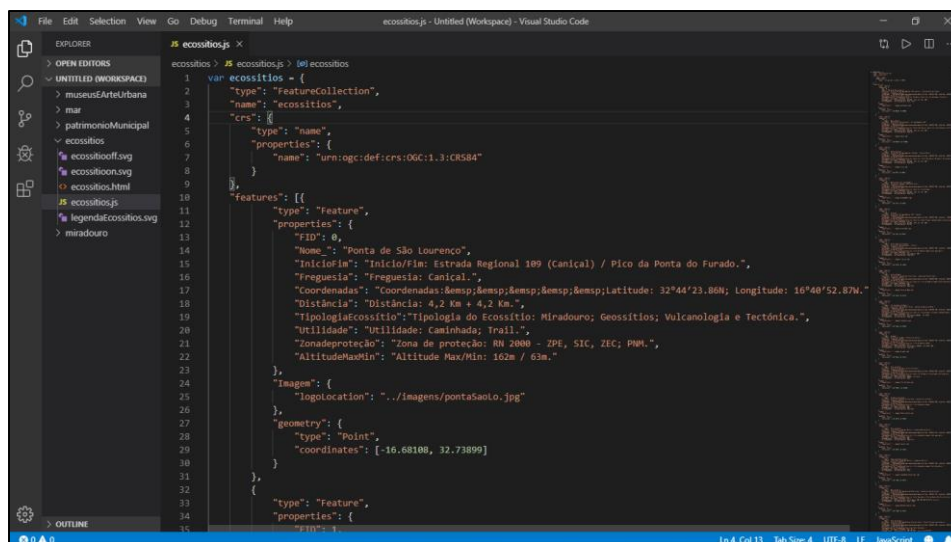


Figura 31: Alterações no GeoJSON.

É possível alterar-se o nome que aparece no toggle, a descrição, a freguesia, a imagem que acompanha a descrição ou as coordenadas onde colocamos o ícone no mapa. Neste script do GeoJSON consegue-se ainda aumentar ou eliminar os campos do toggle. É no GeoJSON que se comanda toda a informação dos pontos de interesse que aparece ao utilizador.

6.2. Integração da aplicação webSIG no *website* VisitMachico

Infelizmente no dia 30 de setembro de 2019, dia da entrega do presente relatório nos serviços académicos ainda não está disponível a aplicação webSIG no *website* VisitMachico. O passo de colocar a aplicação nos servidores do *website* ainda não foi realizado, não obstante todo o *script* estar terminado e o único passo em falta ser colocar o *script* da aplicação nos servidores. A razão pela qual esse passo ainda não foi realizado prende-se com um impasse entre o departamento responsável pelo *website* VisitMachico e a entidade que detém os servidores do *website*.

Embora ainda não esteja disponível no *website*, a posição onde os mapas vão ficar já está definida. Os mapas vão ficar debaixo do tema correspondente no final da página, antes do rodapé do *website*, como podemos ver na Figura 32.

6.2.1. Possíveis alterações da aplicação

As possíveis alterações na aplicação são muitas, visto que está aqui em causa a utilização da linguagem JavaScript que, por sua vez é muito versátil e moldável. Porém as alterações que antevejo, que podem ser realizadas a curto prazo e que fazem sentido no contexto do *website* e do objetivo do mesmo, são enumeradas de seguida.

- Colocar âncoras no *toggle* para que quando se clicar nessas, automaticamente o *browser* conduza para outra parte do *website* ou até para outro *website*, onde exista um maior volume de informação sobre o assunto em pesquisa.
- Aumentar a quantidade de pontos de interesse para que os mapas fiquem com mais informação.
- Dispor novos *plugins* da biblioteca Leaflet, como por exemplo um instrumento de medição.
- Colocar um botão no *toggle* que automaticamente apresenta a rota para o ponto de interesse no *google maps*.
- Utilizar a aplicação para outros temas, ou aplicar no *website* da Câmara Municipal tal como está, uma vez que vem de encontro dos conteúdos do *website*.

6.3. Síntese e conclusões do estágio

Com a realização do estágio adquiri diversas competências, neste momento estou mais preparado para criar e tratar diferentes tipos de dados e de base de dados, tenho uma melhor capacidade para interpretar scripts em JS, HTML e CSS, adquiri capacidade de modificar o front-end de uma aplicação webSIG e estou mais conhecedor das diferentes maneiras de realizar um webSIG.

A realização deste estágio prendeu-se com a necessidade que o município de Machico teve na divulgação dos seus pontos de interesse, para tal foi desenvolvido um webSIG de conteúdos turísticos para a câmara municipal de Machico para dar a conhecer deste modo os pontos de interesse de uma maneira visualmente atraente e fácil de interpretar.

Se a maneira mais comum de valorizar os recursos está na forma como os divulgamos e promovemos junto do público alvo, temos de oferecer ferramentas para este conhecer rapidamente o que mais interessante existe no território, tornando assim a experiência ao local mais dinâmica e eficiente.

Com base neste pressuposto criamos um webSIG que tem um total de cinco mapas com cinco temas diferentes que procurou unir todos os pontos de interesse do município. Para tornar real este webSIG e para que este correspondesse ao que a câmara municipal de

Machico procurava, foi desenvolvido diversos requisitos funcionais e não funcionais, onde no final do projeto conseguimos verificar que foram todos realizados com sucesso.

Ao longo do projeto foi realizada uma triagem dos dados disponibilizados, de modo a que os mapas não sejam demasiado exaustivos visualmente e para que os mesmos só contenham pontos de real interesse turístico.

Todos os dados foram passados para o formato GeoJSON, formato este muito versátil e moldável, de forma a que estes estivessem preparados para ser interpretados por qualquer SIG ou API do mercado.

Outra das premissas que ficou patente no decorrer do estágio foi que um webSIG para fins turísticos pode utilizar na sua construção a linguagem JavaScript com recurso à biblioteca Leaflet. Esta combinação verificou-se ser muito versátil e traz muita flexibilidade na altura de desenvolver o webSIG.

Verificando o resultado, podemos ver que a aplicação é “*user friendly*” e que contém todas as ferramentas para uma navegação simples e eficiente. Outro dos pontos primordiais no que toca a utilização da aplicação, foi esta ser responsiva em qualquer plataforma, tanto no mobile como no formato *desktop*.

Sendo que todas as aplicações necessitam de ser atualizadas para continuar com o mesmo desempenho e com a informação mais recente, foi criado um script simples onde qualquer técnico da câmara consiga proceder a pequenas atualizações nos dados e no conteúdo.

Depois da aplicação estar desenvolvida foi realizado diversos testes de performance para verificar se existia algum erro na aplicação e de maneira a simular a utilização na mesma.

Diferentes abordagens aos webSIGs são possíveis, podemos abordar de uma maneira mais rígida através das arquiteturas aplicacionais existentes ou podemos criar algo do zero completamente à medida do cliente e do tema que vamos abordar através de linguagens de programação. Embora as duas abordagens aos webSIG sejam diferentes o objetivo passará sempre pelo mesmo, que será trazer a melhor informação da forma mais dinâmica e intuitiva a quem dela necessita.

Bibliografia

- Alesheikh, A. A., & Helali, H. (2015). Web GIS: Technologies and Its Applications.
- Bos, B. (2016). A brief history of CSS until 2016. Acedido em: 13 de agosto de 2019, de <https://www.w3.org/Style/CSS20/history.html>
- Caldeira, C. P. (2015). Introdução ao HTML. Universidade de Évora. Évora. Acedido em: 3 de setembro de 2019, de <https://dspace.uevora.pt/rdpc/bitstream/10174/13240/1/Introdu%C3%A7%C3%A3o%20ao%20HTML.pdf>
- CloudMade. (2011). Announcing Leaflet- a Modern Open Source JavaScript Library for Interactive Maps. Acedido em: 7 de setembro de 2019, de <https://archive.is/20140811222442/http://blog.cloudmade.com/2011/05/13/announcing-leaflet-a-modern-open-source-javascript-library-for-interactive-maps/>
- DREM - Direção Regional de Estatística da Madeira. (2017). Análise dos principais resultados definitivos ano de 2016. Acedido em: 12 de julho de 2019, de <https://estatistica.madeira.gov.pt/download-now-3/economic/turismo-gb/turismo-emfoco-gb/tourism-in-focus-gb/finish/288-turismo-em-foco/7843-em-foco-2016.html>
- Duran, E., Shrestha, M., & Seker, D. Z. (2004). Web Based Information System for Tourism Resorts; A Case Study for Side/Manavgat. *Proceedings for XXth International Society for Photogrammetry and Remote Sensing, Istanbul, Turkey: July, (November 2017)*, 12–23.
- Flanagan, D. (2011). *JavaScript: The Definitive Guide, Sixth Edition*.
- Grillo, F. D. N., & Fortes, R. P. de M. (2008). *Aprendendo JavaScript*. Acedido em: 14 de setembro de 2019, de http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_72.pdf
- Khan Academy. (2017). Pausa para a história- como John criou o jQuery? Acedido em: 7 de setembro de 2019, de <https://pt.khanacademy.org/computing/computer-programming/html-js-jquery/jquery-dom-access/a/history-of-jquery>
- Leaflet. (2019). Leaflet - a JavaScript library for interactive maps. Acedido em: 7 de setembro de 2019, de <https://leafletjs.com/blog.html>
- Lebec, G. (2019). JavaScript- A History for Beginners. Acedido em: 4 de agosto de 2019, de <https://www.coursereport.com/blog/history-of-javascript>
- Longman, A. W. (1998). Chapter 2 - A History of HTML. Acedido em: 13 de agosto de 2019, de <https://www.w3.org/People/Raggett/book4/ch02.html>
- Masron, T., Marzuki, A., Mohamed, B., & Ayob, N. M. (2014). Conceptualise Tourism Support System Through Web-Based Gis For Collaborativetourism Planning. *Journal of the Malaysian Institute of Planners, XII*, 59–80.
- Mitchell, T. (2005). *Web Mapping Illustrated*. (S. St.Laurent, Ed.). Sebastopol: O'Reilly Media.

- Oliveira, V. F. C. de. (2015). *Modo de Gestão para a Sustentabilidade de Destinos Turísticos*. Instituto superior técnico.
- Resig, J. (2016). 10th Anniversary of jQuery. Acedido em: 8 de setembro de 2019, de <https://johnresig.com/blog/10th-anniversary-of-jquery/>
- Tutorials Point. (2015). *jQuery web application library*. Tutorials Point (I) Pvt. Ltd. All.
- Valério, D. R. (2014). *Componentes Gráficos em HTML5*. Universidade de Aveiro.
- W3schools. (2019). HTML5 Introduction. Acedido em: 13 de agosto de 2019, de https://www.w3schools.com/html/html5_intro.asp
- York, R. (2015). *Web Development with jQuery*. Indianapolis: John Wiley & Sons, Inc.
- Zeldman, J. (2001). *Taking your talent to the Web : a guide for the transitioning designer*. Kahle/Austin Foundation.

Apêndice A: Script da aplicação (Mapa Ecosystems)

```
<!DOCTYPE html>
<html>

<head>

  <title>Ecosystems</title>

  <meta charset="utf-8" />
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
    integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/1+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUytO7
+7N4QKrDh+drA=="
    crossorigin="" />
  <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
    integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSW0StnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrV
GSmPh6Qw5lwr=="
    crossorigin=""></script>
  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
umd/popper.min.js"
    integrity="sha384-
U02eT0CpHqdsSJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
    integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDs4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
  <link href="https://fonts.googleapis.com/css?family=Bitter|Hind" rel=
"stylesheet">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
trap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossor
igin="anonymous">
  <script src="ecosistios.js" type="text/javascript"></script>
  <script type="text/javascript" src="../leaflet-search.js"></script>
  <link rel="stylesheet" href="../leaflet-search.css" />
  <link rel="stylesheet" href="../designmapa.css" />

</head>

<body>

  <!-- Div do toggle -->
```

```

<div id="toggle" class='out'>
  <div class="close"><button onclick="close_toggle()">x</button></div>
  <h2 id="NomeAttr"> </h2>
  <img id="myImg" src="" alt="Problemas a carregar a imagem." width
  ="230">
  <p id="InicioFim"> </p>
  <p id="Freguesia"> </p>
  <p id="TipologiaEcossítio"> </p>
  <p id="Utilidade"> </p>
  <p id="AltitudeMaxMin"> </p>
  <p id="Zonadeproteção"> </p>
  <p id="Distância"> </p>
  <p id="Coordenadas"> </p>
</div>

<!-- Identificador do mapa -->
<div id="mapid"></div>

<script>

  function close_toggle() {
    $('#toggle').addClass("out");
  }

  // Ligação do meu script ao Identificador e algumas opções do map
  a (como a janela de abertura)
  var mapaCmm = L.map('mapid', {
    minZoom: 10,
    maxZoom: 17, //Maximo tem que ser 17 (0-
18), pois há problemas com um basemap caso seja 18
    zoomControl: false
  }).setView([32.745, -16.78], 13);

  L.control.scale().addTo(mapaCmm);

  // Basemap usados e colocação do selecionador de layer

  var hydda = L.tileLayer('https://{s}.tile.openstreetmap.se/hydda/
full/{z}/{x}/{y}.png', {
    attribution: 'Tiles courtesy of <a href="http://openstreetmap
.se/" target="_blank">OpenStreetMap Sweden</a> &mdash; Map data &copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
  }).addTo(mapaCmm), streets = L.tileLayer('https://server.arcgison
line.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
{

```

```

        attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-
cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-
EGP, and the GIS User Community'
    });

    var baseLayers = {
        "Temático": hydda,
        "Ortofotomapa": streets
    };

    // Defenição dos ícones do mapa

    var ecossitiooff = L.icon({
        iconUrl: 'ecossitiooff.svg',
        iconSize: [48, 58]
    });

    var ecossitioon = L.icon({
        iconUrl: 'ecossitioon.svg',
        iconSize: [48, 58]
    });

    function boldString(str, find) {
        var re = new RegExp(find, 'g');
        return str.replace(re, '<strong>' + find + '</strong>');
    }

    // Colocar os ícones no mapa

    var ecossitio = L.geoJSON(ecossitios);

    function ecossitioLayer(feature,
        layer) {
        layer.setIcon(ecossitiooff);
        layer.layerID = "advanced";

        layer.on("mouseover", function (e) {
            (e.target.setIcon(ecossitioon));
        });

        layer.on("mouseout", function (e) {
            (e.target.setIcon(ecossitiooff));
        });

        layer.on("click", function (e) {
            $('#toggle').removeClass('out');
            $('#description').text(feature.properties.description);
        });
    }

```

```

        $('#NomeAttr').text(feature.properties.Nome_);
        var iniciofim = boldString(feature.properties.InicioFim,
"Inicio/Fim:");
        var freguesia = boldString(feature.properties.Freguesia,
"Freguesia:");
        var Coordenadas = boldString(feature.properties.Coordenad
as, "Coordenadas:");
        var Distância = boldString(feature.properties.Distância,
"Distância:");
        var TipologiaEcoossítio = boldString(feature.properties.Ti
pologiaEcoossítio, "Tipologia do Ecoossítio:");
        var Utilidade = boldString(feature.properties.Utilidade,
"Utilidade:");
        var Zonadeproteção = boldString(feature.properties.Zonade
proteção, "Zona de proteção:");
        var AltitudeMaxMin = boldString(feature.properties.Altitu
deMaxMin, "Altitude Max/Min:");
        $('#InicioFim').html(iniciofim);
        $('#Freguesia').html(freguesia);
        $('#Coordenadas').html(Coordenadas);
        $('#Distância').html(Distância);
        $('#TipologiaEcoossítio').html(TipologiaEcoossítio);
        $('#Utilidade').html(Utilidade);
        $('#Zonadeproteção').html(Zonadeproteção);
        $('#AltitudeMaxMin').html(AltitudeMaxMin);
        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}

L.geoJson(ecossitios, {
    onEachFeature: ecossitioLayer
}).addTo(mapaCmm);

/* ZommControl */
var zoom = new L.Control.Zoom({ position: 'topright' }).addTo(map
aCmm);

// Legenda
var legend = L.control({ position: 'bottomright' });

legend.onAdd = function (map) {
    var div = L.DomUtil.create('div', 'info legend'),
        labels = ["legendaEcossitios.svg"];
    {
        div.innerHTML +=
            " ";
    }
}

```

```

        return div;
    };

    legend.addTo(mapaCmm);

    // Escala
    var escala = L.control.layers(baseLayers).addTo(mapaCmm);

    //SearchBar

    var searchControl = new L.Control.Search({
        layer: ecossitio,
        propertyName: 'Nome_',
        marker: false,
        moveToLocation: function (latlng, title, map) {
            var zoom = map.getBoundsZoom(ecossitio.getBounds());
            map.setView(latlng, 16); // access the zoom
        }
    });

    mapaCmm.addControl(searchControl); //initalize search control

    var marker = ecossitio;
    mapaCmm.removeLayer(marker);

</script>

</body>

</html>

```

Apêndice B: Script da aplicação (Mapa mar)

```

<!DOCTYPE html>
<html>

<head>

    <title>Mar</title>

    <meta charset="utf-8" />
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"

```

```

        integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUyt07
+7N4QKrDh+drA=="
        crossorigin="" />
        <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
        integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSW0StnD0Roefr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrV
GSmPh6Qw5lwrG=="
        crossorigin=""></script>
        <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
umd/popper.min.js"
        integrity="sha384-
U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01clHTMGa3JJDZwrnQq4sF86dIHNDz0W1"
        crossorigin="anonymous"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
        integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
        crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css?family=Bitter|Hind" rel=
"stylesheet">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
trap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T" crossor
igin="anonymous">
        <script src="marinas.js" type="text/javascript"></script>
        <script src="praias.js" type="text/javascript"></script>
        <link rel="stylesheet" href="../designmapa.css" />

</head>

<body>

    <!-- Div do toggle -->

    <div id="toggle" class='out'>
        <div class="close"><button onclick="close_toggle()">x</button></d
iv>

        <h2 id="NomeAttr"> </h2>
        <img id="myImg" src="" alt="Problemas a carregar a imagem." width
="230">
        <p id="description"> </p>
    </div>

    <!-- Identificador do mapa e dimensões -->
    <div id="mapid"></div>

```

```

<script>

    function close_toggle() {
        $('#toggle').addClass("out");
    }

    // Ligação do meu script ao Identificador e algumas opções do map
a (como a janela de abertura)
    var mapaCmm = L.map('mapid', {
        minZoom: 10,
        maxZoom: 17, //Maximo tem que ser 17 (0-
18), pois há problemas com um basemap caso seja 18
        zoomControl: false
    }).setView([32.745, -16.78], 13);

    L.control.scale().addTo(mapaCmm);

    // Basemap usados

    var hydda = L.tileLayer('https://{s}.tile.openstreetmap.se/hydda/
full/{z}/{x}/{y}.png', {
        attribution: 'Tiles courtesy of <a href="http://openstreetmap
.se/" target="_blank">OpenStreetMap Sweden</a> &mdash; Map data &copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
    }).addTo(mapaCmm), streets = L.tileLayer('https://server.arcgison
line.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-
cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-
EGP, and the GIS User Community'
});

    var baseLayers = {
        "Temático": hydda,
        "Ortofotomapa": streets
    };

    // Defenição dos icones do mapa
    var marinaoff = L.icon({
        iconUrl: 'marinaoff.svg',
        iconSize: [48, 58]
    });

    var marinaon = L.icon({
        iconUrl: 'marinaon.svg',

```



```

        iconSize: [48, 58]
    });

    var praiaoff = L.icon({
        iconUrl: 'praiaoff.svg',
        iconSize: [48, 58]
    });

    var praiaon = L.icon({
        iconUrl: 'praiaon.svg',
        iconSize: [48, 58]
    });

    // Colocar os icones no mapa

    var marina = L.geoJSON(marinas);

    function marinaLayer(feature,
        layer) {
        layer.setIcon(marinaoff);
        layer.layerID = "advanced";

        layer.on("mouseover", function (e) {
            (e.target.setIcon(marinaon));
        });

        layer.on("mouseout", function (e) {
            (e.target.setIcon(marinaoff));
        });

        layer.on("click", function (e) {
            $('#toggle').removeClass('out');
            $('#description').text(feature.properties.description);
            $('#NomeAttr').text(feature.properties.Nome_);
            $('#myImg').attr('src', feature.Imagem.logoLocation);
        });
    }

    L.geoJson(marinas, {
        onEachFeature: marinaLayer
    }).addTo(mapaCmm);

    var praia = L.geoJSON(praias);

    function praiaLayer(feature,
        layer) {
        layer.setIcon(praiaoff);
    }

```

```

        layer.layerID = "advanced";

        layer.on("mouseover", function (e) {
            (e.target.setIcon(praiaon));
        });

        layer.on("mouseout", function (e) {
            (e.target.setIcon(praiaoff));
        });

        layer.on("click", function (e) {
            $('#toggle').removeClass('out');
            $('#description').text(feature.properties.description);
            $('#NomeAttr').text(feature.properties.Nome_);
            $('#myImg').attr('src', feature.Imagem.logoLocation);
        });
    }

    L.geoJson(praias, {
        onEachFeature: praiaLayer
    }).addTo(mapaCmm);

    /* ZommControl */
    var zoom = new L.Control.Zoom({ position: 'topright' }).addTo(map
aCmm);

    // Legenda
    var legend = L.control({ position: 'bottomright' });

    legend.onAdd = function (map) {
        var div = L.DomUtil.create('div', 'info legend'),
            labels = ["legendaMar.svg"];
        {
            div.innerHTML +=
                " ";
        }
        return div;
    };

    legend.addTo(mapaCmm);

    // Escala
    var escala = L.control.layers(baseLayers).addTo(mapaCmm);

</script>

```

```
</body>
</html>
```

Apêndice C: Script da aplicação (Mapa miradouros)

```
<!DOCTYPE html>
<html>

<head>

  <title>Miradouros</title>

  <meta charset="utf-8" />
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
    integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWakuyeQUyt07
+7N4QKrDh+drA=="
    crossorigin="" />
  <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
    integrity="sha512-
QVftwZFqvtRNi0ZyCtszn1KSW0StnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrV
GSmPh6Qw5lwr=="
    crossorigin=""></script>
  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
umd/popper.min.js"
    integrity="sha384-
U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01clHTMGa3JJDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
    integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDs4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
  <link href="https://fonts.googleapis.com/css?family=Bitter|Hind" rel=
"stylesheet">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
trap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T" crossor
igin="anonymous">
  <script src="miradouros.js" type="text/javascript"></script>
  <script type="text/javascript" src="../leaflet-search.js"></script>
```

```

<link rel="stylesheet" href="../../leaflet-search.css" />
<link rel="stylesheet" href="../../designmapa.css" />

</head>

<!-- Div do toggle -->

<div id="toggle" class="out">
  <div class="close"><button onclick="close_toggle()">x</button></div>
  <h2 id="NomeAttr"> </h2>
  <img id="myImg" src="" alt="Problemas a carregar a imagem." width="230">
  <p id="description"> </p>
</div>

<!-- Identificador do mapa -->
<div id="mapid"></div>

<script>

  function close_toggle() {
    $('#toggle').addClass("out");
  }

  // Ligação do meu script ao Identificador e algumas opções do mapa (como a janela de abertura)
  var mapaCmm = L.map('mapid', {
    minZoom: 10,
    maxZoom: 17, //Maximo tem que ser 17 (0-18), pois à problemas com um basemap caso seja 18
    zoomControl: false
  }).setView([32.745, -16.75], 13);

  L.control.scale().addTo(mapaCmm);

  // Basemap usados

  var hydda = L.tileLayer('https://{s}.tile.openstreetmap.se/hydda/full/{z}/{x}/{y}.png', {
    attribution: 'Tiles courtesy of <a href="http://openstreetmap.se/" target="_blank">OpenStreetMap Sweden</a> &mdash; Map data &copy; <a href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
  });

```

```

    }).addTo(mapaCmm), streets = L.tileLayer('https://server.arcgison
line.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-
cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-
EGP, and the GIS User Community'
});

var baseLayers = {
    "Temático": hydda,
    "Ortofotomapa": streets
};

// Defenição dos ícones do mapa
var miradourooff = L.icon({
    iconUrl: 'miradourooff.svg',
    iconSize: [48, 58]
});

var miradouroon = L.icon({
    iconUrl: 'miradouroon.svg',
    iconSize: [48, 58]
});

// Colocar os ícones no mapa

var miradouro = L.geoJSON(miradouros);

function miradouroLayer(feature,
    layer) {
    layer.setIcon(miradourooff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(miradouroon));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(miradourooff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}

```

```

    });
}

L.geoJson(miradouros, {
    onEachFeature: miradouroLayer
}).addTo(mapaCmm);

/* ZommControl */
var zoom = new L.Control.Zoom({ position: 'topright' }).addTo(map
aCmm);

// Legenda
var legend = L.control({ position: 'bottomright' });

legend.onAdd = function (map) {
    var div = L.DomUtil.create('div', 'info legend'),
        labels = ["legendaMiradouro.svg"];
    {
        div.innerHTML +=
            " ";
    }
    return div;
};

legend.addTo(mapaCmm);

// Escala
var escala = L.control.layers(baseLayers).addTo(mapaCmm);

// Barra de procura
var searchControl = new L.Control.Search({
    layer: miradouro,
    propertyName: 'Nome_',
    marker: false,
    moveToLocation: function (latlng, title, map) {
        var zoom = map.getBoundsZoom(miradouro.getBounds());
        map.setView(latlng, 16); // zoom de procura
    }
});

mapaCmm.addControl(searchControl); //initialize search control

var marker = miradouro
mapaCmm.removeLayer(marker);

</script>

```

```
</body>
```

```
</html>
```

Apêndice D: Script da aplicação (Mapa museus e arte urbana)

```
<!DOCTYPE html>
<html>

<head>

  <title>Museus e Arte urbana</title>

  <meta charset="utf-8" />
  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
    integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWakuyeQUyt07
+7N4QKrDh+drA=="
    crossorigin="" />
  <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
    integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSW0StnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrV
GSmPh6Qw5lwrg=="
    crossorigin=""></script>
  <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
umd/popper.min.js"
    integrity="sha384-
U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01cllHTMGa3JJDZwrnQq4sF86dIHNDz0W1"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
    integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDs4x0xIM+B07jRM"
    crossorigin="anonymous"></script>
  <link href="https://fonts.googleapis.com/css?family=Bitter|Hind" rel=
"stylesheet">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
trap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQU0hcWr7x9JvoRxT2MZw1T" crossor
igin="anonymous">
  <script src="monumento.js" type="text/javascript"></script>
  <script src="museu.js" type="text/javascript"></script>
```

```

<script src="solar.js" type="text/javascript"></script>
<link rel="stylesheet" href="../designmapa.css" />

</head>

<body>

    <!-- Div do toggle -->

    <div id="toggle" class='out'>
        <div class="close"><button onclick="close_toggle()">x</button></div>
        <h2 id="NomeAttr"> </h2>
        <img id="myImg" src="" alt="Problemas a carregar a imagem." width
        ="230">
        <p id="description"> </p>
    </div>

    <!-- Identificador do mapa e dimensões -->
    <div id="mapid"></div>

    <script>

        function close_toggle() {
            $('#toggle').addClass("out");
        }

        // Ligação do meu script ao Identificador e algumas opções do mapa
        // (como a janela de abertura)
        var mapaCmm = L.map('mapid', {
            minZoom: 10,
            maxZoom: 17, //Maximo tem que ser 17 (0-
            18), pois há problemas com um basemap caso seja 18
            zoomControl: false
        }).setView([32.718400, -16.766667], 16);

        L.control.scale().addTo(mapaCmm);

        // Basemap usados e colocação do selecionador de layer

        var hydda = L.tileLayer('https://{s}.tile.openstreetmap.se/hydda/
        full/{z}/{x}/{y}.png', {
            attribution: 'Tiles courtesy of <a href="http://openstreetmap
        .se/" target="_blank">OpenStreetMap Sweden</a> &mdash; Map data &copy; <a
        href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'

```



```

    }).addTo(mapaCmm), streets = L.tileLayer('https://server.arcgison
line.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-
cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-
EGP, and the GIS User Community'
});

var baseLayers = {
    "Temático": hydda,
    "Ortofotomapa": streets
};

// Defenição dos ícones do mapa
var museuoff = L.icon({
    iconUrl: 'museumoff.svg',
    iconSize: [48, 58]
});

var museuon = L.icon({
    iconUrl: 'museumon.svg',
    iconSize: [48, 58]
});

var solaroff = L.icon({
    iconUrl: 'solaroff.svg',
    iconSize: [48, 58]
});

var solaron = L.icon({
    iconUrl: 'solaron.svg',
    iconSize: [48, 58]
});

var monumentooff = L.icon({
    iconUrl: 'monumentooff.svg',
    iconSize: [48, 58]
});

var monumentoon = L.icon({
    iconUrl: 'monumentoon.svg',
    iconSize: [48, 58]
});

// Colocar os ícones no mapa

var monumentos = L.geoJSON(monumento);

```

```

var museus = L.geoJSON(museu);
var solares = L.geoJSON(solar);

function museuLayer(feature,
    layer) {
    layer.setIcon(museuoff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(museuon));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(museuoff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}

L.geoJson(museu, {
    onEachFeature: museuLayer
}).addTo(mapaCmm);

function solarLayer(feature,
    layer) {
    layer.setIcon(solaroff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(solaron));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(solaroff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
    });
}

```

```

        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}
L.geoJson(solar, {
    onEachFeature: solarLayer
}).addTo(mapaCmm);

function monumentosLayer(feature,
    layer) {
    layer.setIcon(monumentooff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(monumentoon));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(monumentooff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}
L.geoJson(monumento, {
    onEachFeature: monumentosLayer
}).addTo(mapaCmm);

/* ZommControl */
var zoom = new L.Control.Zoom({ position: 'topright' }).addTo(map
aCmm);

// Legenda
var legend = L.control({ position: 'bottomright' });

legend.onAdd = function (map) {
    var div = L.DomUtil.create('div', 'info legend'),
        labels = ["legendaCultura.svg"];
    {
        div.innerHTML +=
            " ";
    }
    return div;
}

```

```

    };

    legend.addTo(mapaCmm);

    // Escala
    var escala = L.control.layers(baseLayers).addTo(mapaCmm);

</script>

</body>

</html>

```

Apêndice E: Script da aplicação (Mapa património municipal)

```

<!DOCTYPE html>
<html>

<head>

    <title>Património Municipal</title>

    <meta charset="utf-8" />
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.4.0/dist/leaflet.css"
        integrity="sha512-
puBpdR07980ZvTTbP4A8Ix/l+A4dHDD0DGqYW6RQ+9jxkRFclaxxQb/SJAWZfWAKuyeQUyt07
+7N4QKrDh+drA=="
        crossorigin="" />
    <script src="https://unpkg.com/leaflet@1.4.0/dist/leaflet.js"
        integrity="sha512-
QVftwZFqvtRNi0ZyCtsznlKSW0StnDORoeFr1enyq5mVL4tmKB3S/EnC3rRJcxCPavG10IcrV
GSmPh6Qw5lwr=="
        crossorigin=""></script>
    <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/
umd/popper.min.js"
        integrity="sha384-
U02eT0CpHqdsJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bo
otstrap.min.js"
        integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"

```

```

        crossorigin="anonymous"></script>
        <link href="https://fonts.googleapis.com/css?family=Bitter|Hind" rel=
"stylesheet">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boots
trap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-
gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossor
igin="anonymous">
        <script src="edificios.js" type="text/javascript"></script>
        <script src="igrejas.js" type="text/javascript"></script>
        <script src="capelas.js" type="text/javascript"></script>
        <link rel="stylesheet" href="../designmapa.css" />

</head>

<body>

    <!-- Div do toggle -->

    <div id="toggle" class="out">
        <div class="close"><button onclick="close_toggle()">x</button></d
iv>
        <h2 id="NomeAttr"> </h2>
        <img id="myImg" src="" alt="Problemas a carregar a imagem." width
="230">
        <p id="description"> </p>
    </div>

    <!-- Identificador do mapa e dimensões -->
    <div id="mapid"></div>

    <script>

        function close_toggle() {
            $('#toggle').addClass("out");
        }

        // Ligação do meu script ao Identificador e algumas opções do map
a (como a janela de abertura)
        var mapaCmm = L.map('mapid', {
            minZoom: 10,
            maxZoom: 17, //Maximo tem que ser 17 (0-
18), pois há problemas com um basemap caso seja 18
            zoomControl: false
        }).setView([32.718400, -16.766667], 16);
    </script>

```

```

L.control.scale().addTo(mapaCmm);

// Basemap usados e colocação do selecionador de layer

var hydda = L.tileLayer('https://{s}.tile.openstreetmap.se/hydda/
full/{z}/{x}/{y}.png', {
    attribution: 'Tiles courtesy of <a href="http://openstreetmap
.se/" target="_blank">OpenStreetMap Sweden</a> &mdash; Map data &copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
}).addTo(mapaCmm), streets = L.tileLayer('https://server.arcgison
line.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}',
{
    attribution: 'Tiles &copy; Esri &mdash; Source: Esri, i-
cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-
EGP, and the GIS User Community'
});

var baseLayers = {
    "Temático": hydda,
    "Ortofotomapa": streets
};

// Defenição dos ícones do mapa
var edificiooff = L.icon({
    iconUrl: 'edificiooff.svg',
    iconSize: [48, 58]
});

var edificioon = L.icon({
    iconUrl: 'edificioon.svg',
    iconSize: [48, 58]
});

var igrejaoff = L.icon({
    iconUrl: 'igrejaoff.svg',
    iconSize: [48, 58]
});

var igrejaon = L.icon({
    iconUrl: 'igrejaon.svg',
    iconSize: [48, 58]
});

var capelaoff = L.icon({
    iconUrl: 'capelaoff.svg',
    iconSize: [38, 48]
});

```

```

var capelaon = L.icon({
    iconUrl: 'capelaon.svg',
    iconSize: [38, 48]
});

// Colocar os icones no mapa

var edificio = L.geoJSON(edificios);

function edificioLayer(feature,
    layer) {
    layer.setIcon(edificiooff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(edificioon));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(edificiooff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
        $('#myImg').attr('src', feature.Imagem.logoLocation);
    });
}

L.geoJson(edificios, {
    onEachFeature: edificioLayer
}).addTo(mapaCmm);

var igreja = L.geoJSON(igrejas);

function igrejaLayer(feature,
    layer) {
    layer.setIcon(igrejaoff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(igrejaon));
    });
}

```

```

        layer.on("mouseout", function (e) {
            (e.target.setIcon(igrejaoff));
        });

        layer.on("click", function (e) {
            $('#toggle').removeClass('out');
            $('#description').text(feature.properties.description);
            $('#NomeAttr').text(feature.properties.Nome_);
            $('#myImg').attr('src', feature.Imagem.logoLocation);

        });
    }

    L.geoJson(igrejas, {
        onEachFeature: igrejaLayer
    }).addTo(mapaCmm);

var capela = L.geoJSON(capelas);

function capelaLayer(feature,
    layer) {
    layer.setIcon(capelaoff);
    layer.layerID = "advanced";

    layer.on("mouseover", function (e) {
        (e.target.setIcon(capelaon));
    });

    layer.on("mouseout", function (e) {
        (e.target.setIcon(capelaoff));
    });

    layer.on("click", function (e) {
        $('#toggle').removeClass('out');
        $('#description').text(feature.properties.description);
        $('#NomeAttr').text(feature.properties.Nome_);
        $('#myImg').attr('src', feature.Imagem.logoLocation);

    });
}

L.geoJson(capelas, {
    onEachFeature: capelaLayer
}).addTo(mapaCmm);

/* ZommControl */

```



```

    var zoom = new L.Control.Zoom({ position: 'topright' }).addTo(map
aCmm);

    // Legenda
    var legend = L.control({ position: 'bottomright' });

    legend.onAdd = function (map) {
        var div = L.DomUtil.create('div', 'info legend'),
            labels = ["legendaPatrimonio.svg"];
        {
            div.innerHTML +=
                " ";
        }
        return div;
    };

    legend.addTo(mapaCmm);

    // Escala
    var escala = L.control.layers(baseLayers).addTo(mapaCmm);

</script>

</body>

</html>

```

Apêndice F: Exemplo de um GeoJSON

```

var praias = {
  "type": "FeatureCollection",
  "name": "praias",
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
    }
  },
  "features": [{
    "type": "Feature",
    "properties": {

```

```

        "Nome_": "Praia da Banda D'Além",
        "description": "Esta praia de areia amarela é uma das poucas referências do género na Ilha da Madeira e está enquadrada na emblemática baía de Machico inserida numa envolvência agradável, tornando-se por isso, num dos locais favoritos a nível regional."
    },
    "Imagem": {
        "logoLocation": "../imagens/alem.jpg"
    },
    "geometry": {
        "type": "Point",
        "coordinates": [-16.762253, 32.718428]
    }
},
{
    "type": "Feature",
    "properties": {
        "Nome_": "Praia de São Roque",
        "description": "A Praia de São Roque, situada na belíssima baía de Machico, é uma praia de calhau rolado remanescente da época dos descobrimentos. Encaixado entre a Capela de São Roque e a foz da Ribeira de Machico, esta praia de acesso livre dispõe das infraestruturas necessárias a uma ida à praia em segurança. Os utentes podem usufruir de serviços vários, como WCs, chuveiros, posto de primeiros socorros, guardas-sóis e na época banhar a praia é vigiada por nadadores salvadores."
    },
    "Imagem": {
        "logoLocation": "../imagens/roquePraia.jpg"
    },
    "geometry": {
        "type": "Point",
        "coordinates": [-16.764669, 32.716240]
    }
},
{
    "type": "Feature",
    "properties": {
        "Nome_": "Praia da Prainha",
        "description": "A Prainha é uma das praias que mais fascina os visitantes pela sua singularidade. É uma pequena praia de areia natural proveniente da formação basáltica, escondida entre as dunas da Piedade, situado nas proximidades da Ponta de São Lourenço. A particularidade desta praia começa logo na acessibilidade pois para lá chegar envolve uma pequena caminhada, com cerca de 100 metros de comprimento, que o conduzirá até à pitoresca baía rodeada por formações geológicas fascinantes."
    },
    "Imagem": {
        "logoLocation": "../imagens/prainha.jpg"
    },

```

```

        "geometry": {
            "type": "Point",
            "coordinates": [-16.715626, 32.742886]
        },
    },
    {
        "type": "Feature",
        "properties": {
            "Nome_": "Praia da Ribeira do Natal",
            "description": "A Praia da Ribeira do Natal, localizada no Caniçal, é banhada por águas amenas e cristalinas galardoada em 2016 com classificação de Ouro pela QUERCUS. Toda a extensão da praia é acompanhada pela promenade do Caniçal e possui infraestruturas como espreguiçadeiras e guarda-sóis fixos, vigilância durante a época balnear, um bar de apoio e parque de estacionamento que aliado ao acesso livre, tornam esta pequena enseada num verdadeiro paraíso."
        },
        "Imagem": {
            "logoLocation": "../imagens/natal.jpg"
        },
        "geometry": {
            "type": "Point",
            "coordinates": [-16.745283, 32.734976]
        }
    },
    {
        "type": "Feature",
        "properties": {
            "Nome_": "Complexo Balnear do Caniçal",
            "description": "Para além das águas cristalinas e amenas do mar, a Vila do Caniçal detém duas piscinas de água salgada localizadas no Complexo Balnear do Caniçal. O Complexo possui acesso ao mar e infraestruturas como um snack-bar, balneários, espreguiçadeiras e chapéus-de-sol."
        },
        "Imagem": {
            "logoLocation": "../imagens/canical.jpg"
        },
        "geometry": {
            "type": "Point",
            "coordinates": [-16.738793, 32.736224]
        }
    },
    {
        "type": "Feature",
        "properties": {
            "Nome_": "Baía de Abra",

```

```

        "description": "A Baía de Abra é uma pitoresca enseada localizada na Ponta de São Lourenço. Utilizado nomeadamente por pescadores, a pequena praia é um misto de calhau e areia preta. A sua intimista localização é a maior atratividade, oferecendo ao banhista o sossego e tranquilidade de uma praia deserta a pouca distância da estrada."
    },
    "Imagem": {
        "logoLocation": "../imagens/abra.jpg"
    },
    "geometry": {
        "type": "Point",
        "coordinates": [-16.696145, 32.747608]
    }
}, {
    "type": "Feature",
    "properties": {
        "Nome_": "Cais do Sardinha",
        "description": "O Caís do Sardinha, situado quase no fim do PR 8 – Vereda da Ponta de São Lourenço, é uma paragem obrigatória onde pode refrescar-se após uma caminhada fascinante pela árida península. Não é uma praia tradicional com amplas acessibilidades e serviços de apoio, mas antes um local com acesso ao mar cristalino. Não é um local com vigilância pelo que deverá ter todas as precauções caso pretenda se refrescar."
    },
    "Imagem": {
        "logoLocation": "../imagens/sardinha.jpg"
    },
    "geometry": {
        "type": "Point",
        "coordinates": [-16.685728, 32.742110]
    }
}, {
    "type": "Feature",
    "properties": {
        "Nome_": "Maiata",
        "description": "Mais do que um local para banhos, a praia da Maiata no Porto da Cruz é um hotspot para a prática do surf, sendo uma das referências a nível regional. Localizada na Maiata de Baixo, a praia em si é um misto de areia preta e calhau rolado e as ondas ligeiras apresentam as condições ideais para os aficionados de surf de qualquer nível. O acesso à praia é livre, contudo não possui infraestruturas de apoio nem em qualquer tipo de vigilância."
    },
    "Imagem": {
        "logoLocation": "../imagens/maiata.jpg"
    },
    "geometry": {
        "type": "Point",

```

```

        "coordinates": [-16.821713, 32.767926]
    }, {
        "type": "Feature",
        "properties": {
            "Nome_": "Complexo Balnear do Porto da Cruz",
            "description": "A pequena vila do Porto da Cruz, para além das praias de areia negra, possui ainda um Complexo Balnear constituído por duas piscinas, sendo uma escolha fantástica para pais e famílias que queiram disfrutar do conforto proporcionado pelos complexos desta natureza. Com infraestruturas de apoio como balneários, sanitários, duches, espreguiçadeiras, guardasóis, e serviços de restauração, o Complexo Balnear está situado no belo passeio marítimo do Porto da Cruz com vista sobre o oceano Atlântico e sobre a imponente encosta."
        },
        "Imagem": {
            "logoLocation": "../imagens/portoDaCruz.jpg"
        },
        "geometry": {
            "type": "Point",
            "coordinates": [-16.827033, 32.773860]
        }
    }, {
        "type": "Feature",
        "properties": {
            "Nome_": "Praia da Alagoa",
            "description": "Encaixada no fundo da imponente Penha D'Águia, a Praia da Alagoa é um pitoresco recanto situado na Vila do Porto da Cruz, considerado um dos melhores pontos para praticar surf e bodyboard na Ilha da Madeira. A Praia da Alagoa com as suas areias negras detém bons acessos ao mar e dispõe de balneários, sanitários, zona de restauração e estacionamento."
        },
        "Imagem": {
            "logoLocation": "../imagens/alagoa.jpg"
        },
        "geometry": {
            "type": "Point",
            "coordinates": [-16.829112, 32.774729]
        }
    }
]
}

```